

**Semantische und logische  
Datenmodellierung multidimensionaler  
Strukturen am Beispiel Microsoft®  
SQL Server™ „Yukon“**

**Diplomarbeit**  
im Fach Informationstechnik  
Studiengang Informationswirtschaft  
der  
Fachhochschule Stuttgart –  
Hochschule der Medien

**Michael Jetter**

Erstprüfer: Prof. Dr.-Ing. Peter Lehmann  
Zweitprüfer: Prof. Dr. Wolf-Fritz Riekert

Bearbeitungszeitraum: 29. April 2004 bis 27. August 2004

Stuttgart, August 2004

## Kurzfassung

Diese Diplomarbeit beschreibt die semantische und logische Datenmodellierung mehrdimensionaler Datenbanken. Aufbauend auf den Grundlagen der Modellierung für DV-Systeme wird die semantische und logische Datenmodellierung erläutert. Die verschiedenen Modellierungsmöglichkeiten und Methoden zum Design einer mehrdimensionalen Datenbank werden hier vorgestellt. Dabei erfolgt ein Vergleich zwischen relationaler und multidimensionaler Datenbankentwicklung. Das Star-Schema und Snowflake-Schema werden als mehrdimensionale Datenmodelle ausführlich erklärt. Die speziellen Modellierungsaspekte von „Zeit“, welche im Data Warehouse eine besondere Rolle besitzt, wird ausführlich dargestellt. Im weiteren Verlauf erfolgt eine Illustration der Data Warehouse Referenzarchitektur und des Microsoft SQL Server 2005. Dabei werden die Business Intelligence Tools und Funktionalitäten des SQL Servers vorgestellt.

Die erläuterten Modelle der gesamten Arbeit werden abschließend anhand einer Fallstudie erarbeitet und nach dem 3-Ebenen Modell semantisch, logisch und physisch auf den SQL ServerExpress 2005 implementiert. Daran kann der Entwicklungszyklus einer mehrdimensionalen Datenbank gezeigt werden.

**Schlagwörter:** mehrdimensionale Datenmodellierung, semantisch, logisch, Entity-Relationship-Modell, Star-Schema, Snowflake-Schema, mehrdimensionale Datenbank, Microsoft SQL Server, Business Intelligence

## Abstract

This diploma thesis discusses the semantically and logically modelling of multidimensional databases. Based on the fundamentals of modelling data processing systems the semantically and logical data models will be explained. Several methods of designing aspects for dimensional modelling are shown. The diploma includes a comparison between relational and dimensional database development. The star-schema and snowflake-schema are introduced, explained and compared. The special aspect of "time" in data warehouses is discussed in detail. The data warehouse architecture and the Microsoft SQL Server architecture are explained in the fourth chapter. The tools for the business intelligence work with the SQL Server are shown.

The designed models of the work are finally shown in a sales case study. Following the 3-level-model these models are worked out semantically, logical and physical subsequently implemented on the SQL ServerExpress 2005. This shows a development cycle of dimensional databases.

**Keywords:** dimensional modelling, semantic, logical, entity-relationship-model, star-schema, snowflake-schema, multidimensional database, Microsoft SQL Server, business intelligence

# Inhaltsverzeichnis

<b>Kurzfassung .....</b>	<b>2</b>
<b>Abstract .....</b>	<b>2</b>
<b>Inhaltsverzeichnis .....</b>	<b>3</b>
<b>Abbildungsverzeichnis .....</b>	<b>5</b>
<b>Tabellenverzeichnis .....</b>	<b>6</b>
<b>Abkürzungsverzeichnis .....</b>	<b>7</b>
<b>1     Einleitung.....</b>	<b>8</b>
1.1   Zielsetzung.....	10
1.2   Aufbau der Arbeit .....	11
<b>2     Datenmodelle und Modellierungsmethoden .....</b>	<b>12</b>
2.1   Modellierungsebenen.....	15
2.1.1   Semantische Modellierungsebene.....	17
2.1.2   Logische Modellierungsebene .....	17
2.1.3   Physische Modellierungsebene .....	18
2.1.4   Zusammenfassung und Fazit.....	19
2.2   Modellarten .....	19
2.2.1   Semantische Datenmodellierung .....	20
2.2.2   Logische Datenmodellierung .....	23
2.2.3   Ziele semantischer und logischer Datenmodellierung .....	24
2.2.4   Zusammenfassung und Fazit.....	25
2.3   Von Fakten und Dimensionen über Fakt- und Dimensionsdaten zum Datenwürfel.....	25
2.4   Vom Entity-Relationship-Modell zum mehrdimensionalen Datenmodell – Ein Vorschlag zur Modellerstellung.....	29
<b>3     Multidimensionale Strukturen für relationale Datenbanken .....</b>	<b>34</b>
3.1   Modellierungsmethoden für relationale Datenbanken .....	39
3.1.1   Star-Schema .....	40
3.1.2   Multistar-Schema .....	44
3.1.3   Snowflake-Schema .....	44
3.1.4   Eine Gegenüberstellung von Star-Schema und Snowflake-Schema.....	47
3.1.5   Zusammenfassung und Fazit.....	48
3.2   Multidimensionale Datenbanken.....	48
3.2.1   On-Line Analytical Processing (OLAP) .....	49
3.2.2   Relationales OLAP (ROLAP) .....	50
3.2.3   Multidimensionales OLAP (MOLAP).....	50

3.2.4	Hybrides OLAP (HOLAP).....	51
3.2.5	Ein Vergleich der OLAP-Architekturen.....	51
3.2.6	Zusammenfassung und Fazit.....	52
3.3	Modellierung von Zeit.....	52
3.4	Vom mehrdimensionalen Entity-Relationship-Modell zur Methode „Hahne“ .....	56
<b>4</b>	<b>Architektur und Arbeitsweise des Microsoft SQL Server 2005 .....</b>	<b>58</b>
4.1	Die Data Warehouse Referenzarchitektur und die Microsoft SQL Server 2005 Architektur .....	58
4.2	Business Intelligence mit SQL Server 2005 – Analysis Services .....	63
4.2.1	Analysis Services Enhancements .....	64
4.2.2	Reporting Services.....	67
4.3	Zusammenfassung .....	68
<b>5</b>	<b>Sales Fallstudie.....</b>	<b>69</b>
5.1	Datenmodelle der Fallstudie .....	70
5.2	Transformation der Datenmodelle in das Datenbankmanagementsystem .....	79
<b>6</b>	<b>Zusammenfassung und Ausblick.....</b>	<b>81</b>
	<b>Glossar.....</b>	<b>82</b>
	<b>Literaturverzeichnis .....</b>	<b>85</b>
	<b>Erklärung .....</b>	<b>91</b>

## Abbildungsverzeichnis

Abbildung 1: Ebenen der Modellierung .....	15
Abbildung 2: Phasen des Datenbank-Entwurfprozesses .....	16
Abbildung 3: Vorgehensmodell für die Erstellung von Data Warehouses.....	19
Abbildung 4: Entity-Relationship-Modell .....	21
Abbildung 5: Multidimensionaler Datenwürfel mit Bestandteilen und Beispieldaten.....	27
Abbildung 6: Mehrdimensionaler Datenwürfel oder Data-Cube der Fallstudie .....	27
Abbildung 7: Multidimensionales Schema der Fallstudie in mE/R-Notation .....	31
Abbildung 8: Dreidimensionaler Daten-Würfel aus der Fallstudie.....	34
Abbildung 9: Dualismus von Würfel und Tabelle .....	35
Abbildung 10: Übergang von ER-Diagramm zum Star-Schema .....	40
Abbildung 11: Beispiel eines Star-Schemas .....	41
Abbildung 12: Multistar-Modell in einer Schemadarstellung .....	44
Abbildung 13: Snowflake-Schema aus der Fallstudie.....	45
Abbildung 14: Zeithierarchie .....	54
Abbildung 15: Dimension mit einem Merkmal und deren Beschreibungen.....	56
Abbildung 16: Modellierung von Basis-Cubes .....	57
Abbildung 17: Referenzarchitektur für ein Data Warehouse-System.....	59
Abbildung 18: Data Warehouse Anwendungen .....	60
Abbildung 19: Schema-Architektur des Microsoft Data Warehouses .....	61
Abbildung 20: SQL Server Speichervarianten .....	64
Abbildung 21: Architektur Analysis Services.....	64
Abbildung 22: Der administrative Zugriff auf den Analysis-Server.....	65
Abbildung 23: Architektur der Reporting Services – SQL Server 2000.....	67
Abbildung 24: Regionen-Aufteilung der Fallstudie .....	69
Abbildung 25: Identifizierte Entitäten der Fallstudie .....	71
Abbildung 26: ER-Modell der Fallstudie – Vertrieb .....	72
Abbildung 27: ER-Modell der Fallstudie – Marketing .....	72
Abbildung 28: Business-Szenario der Fallstudie im Datenmodell.....	73
Abbildung 29: Business-Szenario im Datenmodell mit Attributen .....	73
Abbildung 30: Star-Schema Vertrieb.....	76
Abbildung 31: Star-Schema Marketing .....	76
Abbildung 32: Business-Szenario im Galaxy-Schema.....	77
Abbildung 33: Business-Szenario im Galaxy-Schema nach Vertrieb und Marketing....	77
Abbildung 34: Business-Szenario als mehrdimensionales Datenmodell .....	78
Abbildung 35: Sales-Fallstudie im Star-Schema.....	78
Abbildung 36: Sales-Fallstudie im Partial Snow-Flake Schema .....	79
Abbildung 37: Relationenübersicht mit den ermittelten Beziehungen.....	79
Abbildung 38: Beispielabfrage über mehrere Dimensionen der Fallstudie .....	80

## Tabellenverzeichnis

Tabelle 1: Redundante Datenspeicherung in einer Tabelle .....	37
Tabelle 2: Eigenschaften der Fakten- und Dimensionstabellen .....	38
Tabelle 3: Vergleich von relationalem und multidimensionalem Datenbankentwurf .....	39
Tabelle 4: Star-Schema und Snowflake-Schema im Vergleich .....	47
Tabelle 5: OLAP-Architekturen im Vergleich .....	51
Tabelle 6: Datenbankklassifikation .....	53
Tabelle 7: Beispiel einer Dimensionstabelle der Dimension „Zeit“ .....	54
Tabelle 8: Komponenten-Übersicht SQL Server 2005 .....	62
Tabelle 9: Beschreibung der Entitäten für IBI Technologies .....	70
Tabelle 10: Datentypen und deren Beschreibungen .....	74
Tabelle 11: Entitäten mit den Attributen und deren Beschreibung .....	74

## Abkürzungsverzeichnis

API	Application Programming Interface
BI	Business Intelligence
bzw.	beziehungsweise
DB	Datenbank
DBMS	Database Management System (Datenbankmanagementsystem)
d.h.	das heißt
DTS	Data Transformation Services
DV	Datenverarbeitung
DW	Data Warehouse
ER	Entity/Relationship
ER/M	Entity-Relationship-Modell
etc.	etcetera
ETL	extrahieren, transformieren, laden
HOLAP	Hybrides On-Line Analytical Processing
i.d.R.	in der Regel
i.e.S.	im eigentlichen Sinn
MDBMS	Multidimensionales Datenbankmanagementsystem
MDDM	mehrdimensionales Datenmodell
mE/R	Multidimensionales Entity/Relationship
MOLAP	Multidimensionales On-Line Analytical Processing
NF	Normalform
OLAP	On-Line Analytical Processing
RDBMS	Relationales Datenbankmanagementsystem
ROLAP	Relationales On-Line Analytical Processing
usw.	und so weiter
vgl.	vergleiche
z.B.	zum Beispiel
2NF	Zweite Normalform
3NF	Dritte Normalform

# 1 Einleitung

Analytische Informationssysteme werden für die strategische Unternehmenssteuerung und somit als Informationssystem für Führungskräfte und Entscheider in Unternehmen eingesetzt. In diesem Zusammenhang wird ein Data Warehouse als „ein unternehmensweites Konzept verstanden, dessen Ziel es ist, eine logische zentrale, einheitliche und konsistente Datenbasis für die vielfältigen Anwendungen zur Unterstützung der analytischen Aufgaben von Fach- und Führungskräften aufzubauen, die losgelöst von den operativen Datenbanken betrieben wird.“<sup>1</sup> Aufgrund der Datenanalyse dieser Systeme erfolgen fundierte Entscheidungen, die maßgebliche Auswirkungen auf die Unternehmensstrategie nach sich ziehen. Deshalb sind bereits bei der Systementwicklung und den dabei anzufertigenden Datenmodellen<sup>2</sup> einige Besonderheiten zu beachten. Die multidimensionale Datenanalyse, und somit auch die multidimensionale Datenmodellierung, steht bei analytischen Informationssystemen in einem besonderen Fokus.

Die multidimensionale Sichtweise nimmt bei der Datenanalyse in analytischen Informationssystemen eine wichtige Rolle ein. Durch die unterschiedlichen Sichten auf den Datenwürfel können Kennzahlen ermittelt und Rückschlüsse auf die ermittelten Fakten geschlossen werden. Daraus werden z. B. für das Management Signale ermittelt, welche einen Handlungsbedarf anzeigen.

Analyseorientierte Informationssysteme zielen auf die Unterstützung der dispositiven und strategischen Prozesse in Unternehmen ab und haben eine zeitnahe Versorgung betrieblicher Entscheidungsträger mit relevanten Informationen zu Analysezwecken im Fokus.<sup>3</sup>

Die zentrale Datenbank des Data Warehouse ist der Ort für die Speicherung, Organisation und Repräsentation der verfügbaren Daten. Aus dieser zentralen Rolle heraus kommt der Datenorganisation in der zentralen Datenbank „DIE“ bedeutende Rolle zu. Die Speicherstrukturen, welche durch die Modellkonfiguration festgelegt werden, sind ein entscheidender Faktor über Erfolg und Misserfolg eines Data Warehouse.

Provokativ ausgedrückt „beginnt“ die Modellierung nicht mit dem Zeichnen eines Kästchens, sondern sie basiert auf der allgemeinen Theorie der Begriffsbildung (Begriffsrekonstruktion) mit dem Ziel, ein Modell ingenieurgemäß methodisch konstruieren zu können.<sup>4</sup>

Die Modelle sollen intuitiv sein. Mit diesem Hintergedanken wurde die Bezeichnung der „mehrdimensionalen Datenmodelle“ für eine relativ alte Modellbasis gebildet. Das Bild

---

<sup>1</sup> Chamoni, Gluchowski (1998), S. 13

<sup>2</sup> Ein (Daten-)Modell wird in diesem Zusammenhang häufig auch als (Daten-)Schema bezeichnet.

<sup>3</sup> Hahne (2004a), S. 1

<sup>4</sup> Lehmann (2001), S. 158



des Würfels, welcher als Symbol für die mehrdimensionale Datenanalyse gilt, hat sich in der Zwischenzeit als eingängiges Modell etabliert und ist als solches beim Data Warehouse nicht mehr wegzudenken.

Damit die Entscheidungsträger eines Unternehmens ein neu eingeführtes Data Warehouse überhaupt akzeptieren und langfristig nutzen, sind aufgrund der besonderen Qualität von Managementinformationen hohe Anforderungen an die Datenmodellierung zu stellen.<sup>5</sup>

In dem Bereich der Datenverarbeitung kommt es immer wieder zu Kommunikationsschwierigkeiten zwischen DV-Laien (z.B. Anwender) und DV-Fachleuten (z.B. Datenbankentwickler). Diese werden häufig lediglich durch Missverständnisse oder die vermehrte und übertriebene Verwendung des jeweiligen Fach-Jargons hervorgerufen. Durch unterschiedliche Bildung und Verwendung von Begriffssystemen der einzelnen Fachbereiche kann oftmals kein Gedankenaustausch erfolgen.<sup>6</sup> In der Projektarbeit zur Implementierung von Business Intelligence Lösungen treten häufiger Kommunikationsschwierigkeiten auf. Dies beruht auf verschiedenen Gründen. Zum einen auf verschiedenen Interpretationen der zwischenmenschlichen Kommunikation, zum anderen auch auf der Darstellungsweise der zu modellierenden Aspekte.

Gerade im Bereich der Datenmodellierung ist das Kommunikationsproblem so gering wie möglich zu gestalten. Durch die Gestaltung von intuitiven Modellen, die Standardisierung der Modelle und Vereinheitlichung der Modellsprachen und Anwendungen kann dem Missverständnis durch Verwendung unterschiedlichen Vokabulars der Fachgruppen entgegen gewirkt werden. Ein Datenmodell ist durch Vereinfachungen und die Reduzierung auf Fakten, im Vergleich zu dem Ausschnitt der Realwelt, von DV-Anwendern genauso nachvollziehbar wie für die Spezialistengruppe. Dadurch kann ein weitgehend gleiches Verständnis über das zu realisierende System, vorausgesetzt werden.

Beim Aufbau eines Data Warehouse kommt es insbesondere darauf an, die relevanten Begriffe – Kennzahlen und Dimensionen – herauszuarbeiten und das Data Warehouse entsprechend den Anforderungen der Endanwender zu modellieren.<sup>7</sup>

Zum Zeitpunkt der Anfertigung dieser Arbeit steht der Microsoft SQL Server 2005 Beta 2 zur Verfügung. Hierbei handelt es sich um eine SQL ServerExpress-Version, welche nicht alle Funktionen der Endversion des Microsoft SQL Server 2005 zur Verfügung stellt. Bei dieser Beta 2 Version sind keine Analyse-Funktionalitäten enthalten.

---

<sup>5</sup> Mucksch, Behme (2000), S. 42

<sup>6</sup> vgl. Lehner (1995b), S. 73ff.

<sup>7</sup> Hahne (1999), S. 145

## 1.1 Zielsetzung

Ziel der Arbeit ist die Datenmodellierung im Bereich Data Warehouse darzustellen, zu erläutern und anzuwenden. Durch die Anwendung der Datenmodelle auf die Fallstudie und die Anwendung am Beispiel MS SQL Server 2005 soll der Bezug zur Praxis innerhalb der Diplomarbeit hergestellt sein.

Auf die verschiedenen Modellierungsmöglichkeiten, auf die Vorgehensweise bei der Datenmodellierung zur Wahrung der Datenqualität wird im Rahmen dieser Arbeit eingegangen.

Ein durchgängiges Beispiel dient der Verdeutlichung zu den Erläuterungen dieser Arbeit.

Bei jeder Modellerstellung muss darauf geachtet werden, das Modell so einfach wie möglich aufzubauen. Selbstverständlich müssen trotz diesem Aspekt alle relevanten Informationen in dem Modell dargestellt sein. Bei so komplexen Modellen wie im Bereich der Informationssysteme muss auf eine möglichst einfache Darstellung geachtet werden. Bei den einzelnen Entwicklungsschritten zu einem konkreten Data Warehouse werden verschiedene Modellierungstechniken angewandt. Diese spezifizieren in Schritten die Anforderungsanalyse immer weiter, bis das Informationssystem im Betrieb angewendet wird. Informationsverluste dürfen bei dieser Systementwicklung nicht auftreten, da sonst Analyseeinschränkungen für den Anwender entstehen. Eine intuitive und konsistente Systementwicklung muss zu diesen Anforderungen einen großen Beitrag leisten.

Ausgehend von der Anforderungsanalyse kann ein Data Warehouse durch verschiedene Modelle erstellt werden. Wie oben bereits erwähnt, ist dabei auf eine intuitive Darstellung zu achten. Aus der Anforderungsanalyse wird ein semantisches Datenmodell gebildet. Dieses wird zum logischen Modell transformiert um schließlich in das konkrete physische Modell übertragen und implementiert zu werden. Der Schwerpunkt dieser Arbeit liegt auf der semantischen und logischen Datenmodellierung. Dabei wird der Datenmodellierung mit dem Star- und Snowflake-Schema besondere Beachtung gewidmet.

Ein Data Warehouse-Projekt kann dann als erfolgreich abgeschlossen betrachtet werden, wenn das System für verschiedene Anwender maßgeblich zur Informationsbereitstellung relevanter Informationen zur Unternehmensteuerung beiträgt und dieses Unternehmen durch die Informationen einen Kostenvorteil generieren kann.

## 1.2 Aufbau der Arbeit

- Kapitel 1: Im Rahmen der vorliegenden Arbeit werden **multidimensionale Datenmodelle und Modellierungsmethoden** erläutert. Dieses Kapitel wird die Fragestellungen „Was sind Datenmodelle?, „Wozu dienen Datenmodelle?, Wozu dienen Softwaremodelle?“ beantworten. Die Konzepte in der Softwareentwicklung werden unter dem Aspekt „Wie ist ein Softwaremodell aufgebaut?“ betrachtet.
- Kapitel 2: In **Multidimensionale Strukturen für relationale Datenbanken** wird das Modell der Multidimensionalität betrachtet. Es geht in dem Kapitel um das Bild des Würfels und die Darstellungsform in Relationen. Die Modellierung von Zeit ist nicht im Data Warehouse nicht trivial. Worin liegen die Problemstellungen?
- Kapitel 3: Das Kapitel **Multidimensionale Strukturen für Relationale Datenbanken** erläutert die Übertragung der Datenmodelle in die Praxis.
- Kapitel 4: Anschließend wird der neue Microsoft<sup>8</sup> SQL Server 2005 (Projektname „Yukon“) im Rahmen der Arbeit analysiert. **Architektur und Arbeitsweise des Microsoft SQL Server 2005** stellt einen Vergleich zu der DW-Referenzarchitektur dar und erläutert die Funktionsweisen der Business Intelligence Tools.
- Kapitel 5: Die erarbeitete und dargestellte Theorie der Datenmodelle werden im Rahmen einer **Fallstudie** auf den SQL ServerExpress 2005 übertragen.

---

<sup>8</sup> Bei Microsoft, SQL Server 2005 („Yukon“) handelt es sich entweder um eingetragene Trademarks oder ein Trademark von Microsoft Corporation. Anschrift: Microsoft Deutschland GmbH, Konrad-Zuse-Str. 1, 85716 Unterschleißheim, <http://www.microsoft.com/germany>. Microsoft ist nicht Herausgeber dieser Arbeit und ist damit auch nicht für den Inhalt verantwortlich.

## 2 Datenmodelle und Modellierungsmethoden

Der konventionelle Weg der Systementwicklung besteht darin, [...] entwicklungsrelevante Realitätsausschnitte gegebenenfalls gemeinsam mit der Fachabteilung abzubilden. Die für die Darstellung verwendete Sprache wird durch die Modellierungsmethode vorgegeben.<sup>9</sup>

**Modell** wird häufig auf das italienische Wort “modello” zurückgeführt, dies bedeutet „Muster“ oder „Vorbild“. Ein Modell kann wie folgt definiert werden:<sup>10</sup>

- **Wirtschaftsinformatik:** Ein Modell ist ein System, das ein anderes System zielgerichtet abbildet. (Ferstl/Sinz 1993)
- **Informatik:** Ein Modell ist eine idealisierte, vereinfachte, in gewisser Hinsicht ähnliche Darstellung eines Gegenstands, Systems oder sonstigen Weltausschnitts mit dem Ziel, daran bestimmte Eigenschaften des Vorbilds besser studieren zu können. (Hesse et al. 1992)

Traditionellerweise versteht man unter einem **Modell** die **Abbildung der Realität** oder eines Realitätsausschnitts. Die **Modellbildung** ist also ein **Abbildungsprozess**. Eine zentrale Eigenschaft spielt bei diesem Prozess die **Ähnlichkeit**.<sup>11</sup>

Das Kürzel „Modell“ wird häufig synonym zu Konzept, Schema, Plan, Struktur oder Konzeption verwendet. In der Informationstechnologie wird das Kürzel i.d.R. für ein konkretes Datenmodell oder ein Architekturmodell verwendet.

Die Basis für die Entwicklung eines Softwaresystems bildet die Modellierung eines Wirklichkeitsausschnitts. Der darzustellende Wirklichkeitsausschnitt muss zunächst identifiziert und modelliert werden, bevor ein adäquates Informationssystem erstellt werden kann. Der Schlüssel bei der Datenmodellierung liegt darin, den Wirklichkeitsausschnitt möglichst genau zu erfassen und in geeigneter Weise abzubilden, d.h. den Ausschnitt korrekt zu beschreiben. Die gewählte Darstellung sollte möglichst Redundanzfrei sein. Als Ergebnis der Datenmodellierung entsteht ein Datenmodell in Form einer graphischen Darstellung aller Tabellen (Relationen) einer Datenbank, inklusive aller Beziehungen und Schlüssel der Tabellen. Somit entsteht aus dem Datenmodell ein Bild des Datenbankschemas. Dieses Bild muss sich an der Arbeits- und Denkweise der Anwender orientieren.

Es gibt Datenmodelle als Beschreibungssprache und Datenmodelle als Ergebnis der Modellierung. Unter der Gruppe der Datenbeschreibungssprache nennt Date auch das

---

<sup>9</sup> vgl. Lehner (1995b), S. 73ff.

<sup>10</sup> Lehner (1995a), S. 27

<sup>11</sup> Lehner (1995a), S. 27

relationale Modell.<sup>12</sup> In dieser Arbeit werden vorrangig die Datenmodelle als „Datenbeschreibungssprache“ untersucht.

Um für ein DV-Projekt Informationen und Detailangaben zu sammeln und in die verschiedenen Stufen der Entwürfe zu übertragen, können in dem jeweils entsprechenden Datenmodell die Datenklassen (Datenkategorien) festgelegt werden und die Beziehungen zueinander hergestellt werden. Das festlegen der Datenklassen, oder synonym der Entitätsmengen, und Beziehungen werden in dieser Stufe völlig unabhängig von dem später zu verwendenden DV-System vorgenommen. Unter dem DV-System ist in diesem Zusammenhang die Rechnerarchitektur und das Datenbanksystem zu verstehen. Die hier beschriebene Modellierung erfolgt jeweils in drei Schritten. Auf die erste Datenanalyse erfolgt die Festlegung der Entitätsmengen mit dem Entwurf des Entitäten-Beziehungsmodells. Anschließend wird dieses Modell in das konkrete Datenbankschema überführt. Auf die technische Realisierbarkeit wird hier noch nicht geachtet.<sup>13</sup> Die Modellierungsebenen und Methoden der Modellübertragung werden in den Kapiteln 2.1 bis 2.4 näher beschrieben.

Im Rahmen der Datenmodellierung treten stets zwei konkurrierende Zielsetzungen auf:<sup>14</sup>

- a) Das Datenmodell stellt die Grundlage für Diskussionen mit dem Entscheidungsträger dar. Daher muss das Datenmodell die tatsächlichen Zusammenhänge des Verantwortungsbereichs eines Entscheidungsträgers möglichst vollständig und verständlich abbilden, d.h., es muss die Semantik des Realitätsausschnitts wiedergeben. Zu betonen ist, dass das Datenmodell einfach und leicht überschaubar in der Darstellung sein muss, damit ein Entscheidungsträger eines beliebigen Funktionsbereichs im Unternehmen als DV-Laie die im Modell abgebildeten Zusammenhänge schnell nachvollziehen kann.
- b) Darüber hinaus stellt das Datenmodell für den Datenbankdesigner die Grundlage zur Überführung in ein Datenbankmodell dar. Bezüglich seiner Arbeit muss das Datenmodell möglichst viele Detailinformationen zu den Datenstrukturen wiedergeben, die auch für das Erzielen einer optimalen Systemleistung wichtig sind.

Aufgrund der unterschiedlichen Betrachtungen ist es notwendig, einzelne Modellbausteine, je nach Detaillierungsgrad der gewünschten Betrachtung, aus- und einblenden zu können. Diese Modellierungen müssen Plattform- und Hardwareunabhängig durchgeführt werden, um spätere Änderungen mit möglichst wenig anzupassenden Schnittstellen gewährleisten zu können.<sup>15</sup>

---

<sup>12</sup> siehe Date (1986) oder Chen, Knöll (1991)

<sup>13</sup> vgl. Meier (1998), S. 13

<sup>14</sup> Holthuis (2001), S. 117f.

<sup>15</sup> vgl. Holthuis (2000), S. 159

Ein Datenmodell ist das zentrale Hilfsmittel der „Datenbank-Technologie“ zur Herstellung einer Abstraktion von einem gegebenen „Realwelt-Ausschnitt“ und gleichzeitig von den Einzelheiten der physischen Speicherung.<sup>16</sup> Ein Datenmodell bildet somit auch das zentrale Kernstück bei der Datenmodellierung für jedes Data Warehouse. Data Warehouse-Systeme werden i.d.R. in Form von multidimensionalen Datenbanken realisiert. Aus diesem Hintergrund wird hier ein Schwerpunkt auf die Multidimensionalität gelegt.

Ein gutes Datenmodell für multidimensionale Datenmodelle sollte unter anderem die folgenden Anforderungen erfüllen:<sup>17</sup>

- Einbindung der Anwender in die ersten Schritte der Modellierungsprozesse
- Durchgängige Modellierung (→ semantisch → logisch → physisch)
- Verbindung der Modellierungsebenen → Transformationsmöglichkeiten (*Durchgängige Modellierung*)
- Einheitliches Modellierungsschema der Datenbank
- Modell-Dokumentation in Analogie zum Schema-Entwurf
- System muss erweiterbar und in vertretbarem Aufwand wartbar sein
- Zukunftsorientierten Systemaufbau besitzen
- Einheitliche Namenskonventionen der Begriffe (Kennzahlen und Attribute)

„Datenmodelle sind, wie die Anschauungs- und Idealmodelle der Wirtschafts- und Gesellschaftswissenschaften, Beschreibungen. Sie beschreiben jedoch keine Wirklichkeit, sondern ein Wissen über die lebensweltliche Bedeutung (Semantik) sowie über die maschinelle Repräsentation und Manipulation von Daten“.<sup>18</sup>

Die Analyse von komplexen Objekten, Vorgängen, Entwicklungen etc., deren Struktur oder Verhalten im einzelnen noch nicht bekannt ist, geschieht zweckmäßigerweise mit Hilfe eines Modells; dieses kann dann sukzessive um bekannte Faktoren, Strukturen und kausale Zusammenhänge erweitert werden, bis das Modell eine gute Annäherung an die gebildete Realität darstellt.<sup>19</sup>

Ein Datenmodell soll die Bedeutung und Repräsentation von Daten beschreiben.<sup>20</sup> Ziel der Datenmodellierung ist den Ausschnitt der realen Welt, welcher auf dem Informationssystem abgebildet werden soll, schrittweise zu spezifizieren und schließlich möglichst ohne Informationsverlust auf das Datenbankmanagementsystem zu übertragen. Ein Modell bildet dabei nur die jeweils tatsächlich relevanten Aspekte schematisch ab.

---

<sup>16</sup> Vossen (1999), S. 22

<sup>17</sup> vgl. Hahne (2002a), S. 8ff. sowie Holthuis (2001), S. 114ff.

<sup>18</sup> Wedekind, (2001), S. 143

<sup>19</sup> Lehner (1995b), S. 79

<sup>20</sup> Hahne (2002b), S. 401

Dies ist zur Reduzierung der Komplexität eines Modells und zur besseren Übersichtlichkeit für den Betrachter unbedingt erforderlich.

Die zu erwartenden Abfragen der späteren Anwender dienen als Grundlage für die Modellierung im Bereich des Data Warehouse. Von dieser Grundüberlegung sollten alle Modell-Überlegungen bei der Einführung eines Data Warehouse geleitet sein.

Bei der Analyse des Informationsbedarfs wird festgelegt, welche Informationen für das geplante Datenbanksystem von Bedeutung sind. Diese Ergebnisse werden im konzeptionellen Schema durch ein semantisches Datenmodell formalisiert.<sup>21</sup>

Jeder Entwurf für ein Data Warehouse beginnt Idealerweise mit der konzeptuellen Datenmodellierung und den zugehörigen Vorüberlegungen. Bei dieser Modellierungsstufe wird auf die technische Realisierbarkeit nicht geachtet. Dieser Aspekt erleichtert und verkürzt den Modellentwurf erheblich. Die bereits erwähnte Kommunikation der IT-Experten und Domänenexperten wird durch die grafisch orientierten Modelle erleichtert. Durch diese Entity-Relationship-Modelle, stellt der Experte seine „*Sicht auf die Welt*“ in grafischer Form dar.<sup>22</sup> Die zunächst informelle Problembeschreibung wird zur grafischen Darstellung auf ein ER-Modell übertragen. Auf die einzelnen Modellierungsmethoden wird in Kapitel 3.1 näher eingegangen.

## 2.1 Modellierungsebenen

Datenmodelle beschreiben die Bedeutung und die Darstellung von Daten. In diesem Kapitel werden die drei Modellierungsebenen erläutert.

In der Abbildung 1 ist eine geläufige Strukturierung der Modellierungsebenen grafisch dargestellt. Bei den verschiedenen Modellierungsschritten wird ausgehend von der Informationsbedarfsanalyse, jedes Modell weiter verfeinert und so schrittweise zum Zielsystem hin weiter spezifiziert. Jede der Abstraktionsebenen bleibt dabei unabhängig von dem darunter oder darüber liegenden Modell. Dadurch bleibt die Datenunabhängigkeit der verschiedenen Ebenen gewährleistet.

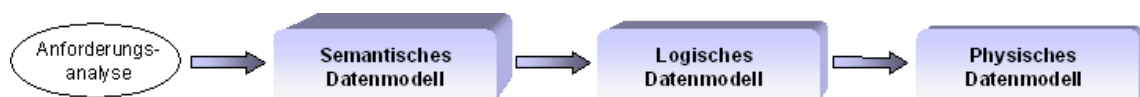


Abbildung 1: Ebenen der Modellierung<sup>23</sup>

Der Entwurfsprozess eines Data Warehouse kann danach in die drei Phasen semantische<sup>24</sup>, logische und physische Modellierungsebene des Entwurfsprozesses unterglie-

<sup>21</sup> Hahne (2002a), S. 11

<sup>22</sup> vgl. Bauer, Günzel (2001), S. 157

<sup>23</sup> Hahne (1999), S. 147

<sup>24</sup> semantisch wird oft auch als konzeptionell oder konzeptual bezeichnet

dert werden. Die graphische Repräsentation beim semantischen Schema ist frei wählbar. Hier gibt es verschiedene Varianten das Schema darzustellen. Das Augenmerk sollte bei dem Modell jedoch eher auf eine übersichtliche Präsentation, als auf den spezifischen Modellcharakter gelegt werden. Aus dem Schema muss ersichtlich sein, welche Anforderungen an das System gestellt werden. In welcher Notation oder Modellsprache dies geschieht ist auf dieser plattformunabhängigen Ebene noch trivial. Ausgehend vom semantischen Datenmodell wird ein logisches Datenmodell erstellt. Dieses Modell ist bereits auf die Anforderungen und Anwendungsstruktur des Informationssystems auszurichten. Je nach Modellierungsanforderungen der darzustellenden Strukturen ist eine geeignete Datenmodellierungssprache für das System zu wählen. Die Gesamtheit dieser Modellierungs-Ebenen und des Entwurf-Prozesses wird in der Abbildung 2 als Grafik dargestellt und.

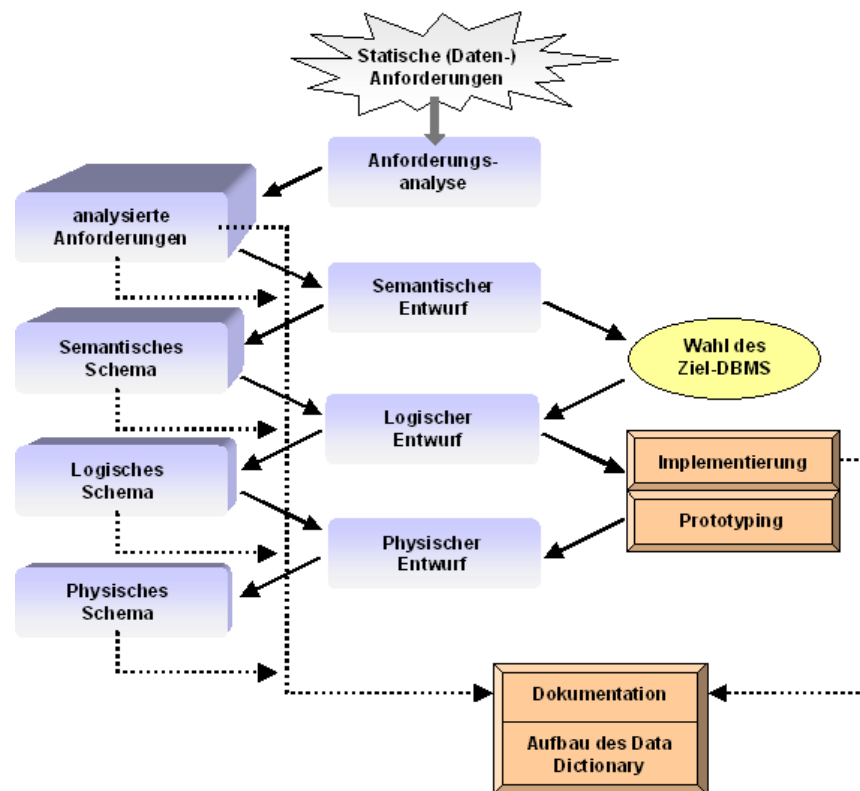


Abbildung 2: Phasen des Datenbank-Entwurfprozesses<sup>25</sup>

### Anforderungsanalyse – Fachkonzept

Ziel der Informationsbedarfsanalyse ist eine möglichst exakte Beschreibung der zu erwartenden Fragestellungen, die an das zu entwickelnde System gerichtet werden.<sup>26</sup> Ausgehend von diesen Fragestellungen muss das Data Warehouse möglichst nahe an den Vorstellungen der Fachbereiche modelliert werden.

<sup>25</sup> Vossen (1999), S. 73

<sup>26</sup> Hahne (2002a), S. 28



### 2.1.1 Semantische Modellierungsebene

Die semantische Modellierungsebene ist der Realwelt am Nächsten. Dadurch werden besondere Anforderungen an die Modellierung dieser Ebene gestellt.

Die Zeichen oder Symbole sollen „Aussagen über Phänomene der realen Welt treffen“.<sup>27</sup> Um den Zeichen und Symbolen eine Bedeutung zuordnen zu können, müssen sie an bestimmte Bedeutungen geknüpft sein. Dadurch werden sie für den Menschen interpretierbar. Betrachtet man z.B. die Ziffernfolge 0711, so handelt es sich scheinbar um eine zufällige Ziffernkombination ohne weiteren Informationsgehalt. Werden diese Ziffern jedoch in dieser Reihenfolge mit der Telekommunikation in Verbindung gebracht, so ist die Vorwahl des Stuttgarter Telefonnetzes in dieser Ziffernkombination hinterlegt. Für den menschlichen Betrachter erhalten die Ziffern erst durch diese semantische Zuordnung eine gewünschte Aussagefähigkeit.

Das Fachkonzept bildet die Grundlage für das semantische Datenmodell.

Im Fachkonzept wird die Bedeutung der handelnden Personen, ihrer Verpflichtungen, Rollen, Handlungen und Mitteilungen beschrieben. Semantische Datenmodelle erlauben nun, die relevanten Objekte der realen Welt mit wenig Informationsverlust abzubilden. Das semantische Schema bildet damit auch die Schnittstelle zu den Benutzern eines Systems. Daher muss auf dieser Ebene aus Sicht des Anwenders definiert werden, welche Kennzahlen und welche Entscheidungsobjekte relevant sind und welche Zusammenhänge zwischen ihnen existieren.<sup>28</sup>

Die *semantische Ebene* beschreibt also Objekte (Entitäten), die im Data Warehouse behandelt werden, unabhängig von der logischen Repräsentation und ihrem physischen Speicher.<sup>29</sup> Das Ziel der semantischen Datenmodellierung ist ein (formales) Abbild des darzustellenden realen Weltausschnitts. Dies soll in formalen Ausdrücken dargestellt werden, um die Betrachtung und Interpretation des Modells zu vereinfachen.

Spätestens nach Abschluss der semantischen Datenmodellierung muss die Entscheidung über das zu verwendende Datenbank-System erfolgen (vgl. Abbildung 2), da alle weiteren Modellierungsschritte und Überlegungen systembezogen vorgenommen werden.<sup>30</sup> Das Ergebnis wird als konzeptuelle Grundlage zum Datenbank-Entwurf verwendet.

### 2.1.2 Logische Modellierungsebene

Das erstellte Datenmodell der semantischen Ebene ist die Grundlage der logischen Modellierungsebene. Das erarbeitete Datenmodell wird durch logische Datenmodelle, in Form eines konkreten Datenmodells aufgearbeitet und somit in die gewählte Modellsprache übertragen. Dabei erfolgt eine „Übersetzung“ der abstrakten Sprache in die

---

<sup>27</sup> Holthuis (2000), S. 183

<sup>28</sup> Totok (2000a), S. 99

<sup>29</sup> Lehmann (2001), S. 89

<sup>30</sup> vgl. Vossen (1999), S. 76

Begriffe des Datenbankmodells. Die Regeln der „Übersetzung“ werden in Metadaten hinterlegt, damit die Transformation von Ebene zu Ebene nachvollziehbar bleibt. Das semantische Datenmodell wird in dieser Designphase präzisiert.

In Abhängigkeit von dem zu verwendenden Datenbanksystem wird das konzeptionelle Modell in das zugrunde liegende Datenbankmodell übertragen. Dieses logische Modell stellt das Bindeglied zwischen konzeptionellem und physischem Design dar.<sup>31</sup> Das entstandene Modell wird als Datenbank-Schema bezeichnet.

### 2.1.3 Physische Modellierungsebene

Auf physischer Ebene werden die Objekte der logischen Ebene umgesetzt bzw. auf einem konkreten Datenbankmanagementsystem (DBMS) implementiert. Hierbei spielen Zugriffsgeschwindigkeit und Speicherplatzoptimierung eine wesentliche Rolle.<sup>32</sup> Beim physischen Entwurf ist ein Design für möglichst kurze Zugriffszeiten auf die zu speichernden Daten zu erstellen. Durch geeignete Speicherstrukturen und Dateisysteme kann das Laufzeitverhalten der zu erwartenden Anfragen deutlich reduziert werden. Daraus entstehen bessere Performanceergebnisse, was den Erfolg jedes Datenbankentwurfs nachhaltig steigert.

Durch das physische Modell werden unter anderem festgelegt:<sup>33</sup>

- Verwendbare Dateiformate
- Zuweisung des Speicherplatzes
- Gruppierung von Blöcken zu Clustern
- Denormalisierung der Tabellen
- Indexauswahl
- Etc.

### Vorgehen bei der Entwicklung von multidimensionalen Informationssystemen

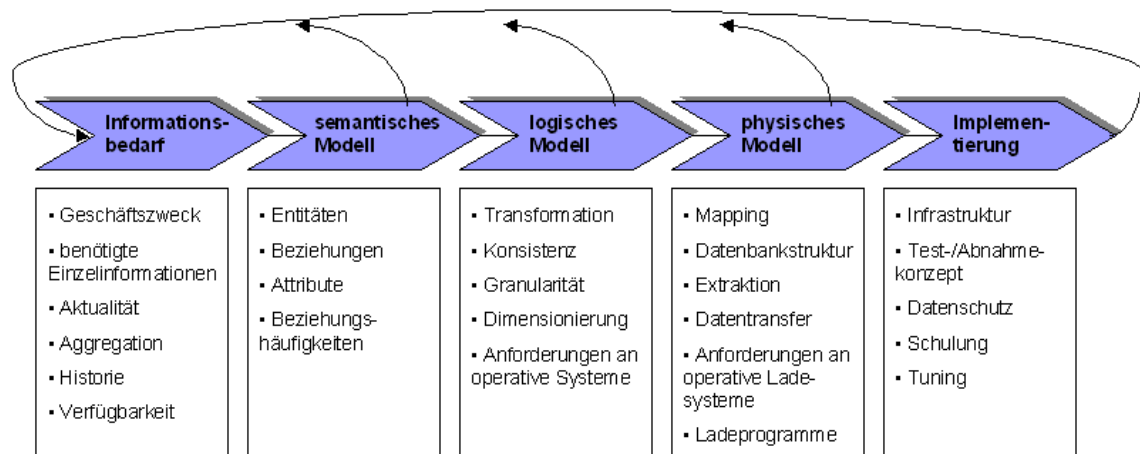
Das Vorgehen, wie in Abbildung 3 dargestellt, entspricht der klassischen Erstellung eines Data Warehouses. Hier ist auch dargestellt, dass bei dem jeweiligen Modell-Entwurf kein völlig autarker Prozess vorliegen muss, sondern dass bei Bedarf das jeweils vorige Modell der Prozesskette anzupassen ist.

---

<sup>31</sup> Hahne (2002a), S. 11

<sup>32</sup> Totok (2000a), S. 99

<sup>33</sup> vgl. Vossen (1999), S. 78

Abbildung 3: Vorgehensmodell für die Erstellung von Data Warehouses<sup>34</sup>

### 2.1.4 Zusammenfassung und Fazit

Der Modellierungsprozess beginnt mit der Problemspezifikation auf einer zunächst zielsystemunabhängigen Ebene und endet als implementierbares Datenbankschema.<sup>35</sup>

Primäres Ziel ist die Bereinigung und allgemeine Konsensfindung der rekonstruierten Fachbegriffe eines Anwendungsgebietes für die fachliche Spezifikation der Systemlösung. Am Ende dieses Teil-Prozesses wird ein semantisches Datenmodell konstruiert, das die relevanten Sachverhalte der realen Welt ohne Informationsverlust abbildet. Das semantische Schema bildet damit auch die Schnittstelle zu den Benutzern eines Systems. Daher muss auf dieser semantischen Ebene aus Sicht des Anwenders definiert werden, welche Kennzahlen und welche Entscheidungsobjekte relevant sind und welcher Zusammenhang zwischen ihnen existiert.<sup>36</sup>

## 2.2 Modellarten

Ein Datenmodell, welches auf einer relativ einfachen Struktur beruht, besitzt letztlich auch die Chance einen komplexen Realitätsausschnitt noch übersichtlich darstellen zu können. Von einer Überfrachtung des Grundmusters der Modellierungssegmente (Symbole, Farbgestaltung, Ebenen, etc.) ist Abstand zu wahren.<sup>37</sup> „Start small think big“ gibt den Gedanken in einfachen Worten wieder. Die Modellgestaltung sollte so einfach wie möglich erfolgen, es muss aber jeder Aspekt bedacht werden und somit auch im Modell darstellbar sein.

<sup>34</sup> in Anlehnung an Totok (2000a), S. 116

<sup>35</sup> Holthuis (2000), S. 186

<sup>36</sup> Lehmann (2001), S. 27

<sup>37</sup> vgl. Kimball (1996), S. 10

In den letzten Jahren ist die Forderung nach einfacher Verständlichkeit, Anwendbarkeit und Benutzbarkeit des in Form von Daten repräsentierten und manipulierten Wissens ins Zentrum der Theorie und Praxis von Datenmodellen gerückt: Es wurden *semantische Datenmodelle* und *konstruktive Verfahren* entwickelt, mit denen sich Wissen über die Bedeutung komplexer Datenbestände explizit repräsentieren lässt.<sup>38</sup>

Eine grafische Notation für multidimensionale Informationssysteme muss in der Lage sein, die Basiskonstrukte, wie Kennzahlen, Dimensionen oder Ableitungsregeln, adäquat in einem Modell abzubilden.<sup>39</sup>

Ergebnis des Entwurfsprozesses ist zunächst das multidimensionale Datenmodell, das Unternehmenszahlen (z.B. Umsätze) sowie Analysemerkmale für diese Kennzahlen (z.B. Umsätze nach Region und Kunden) dokumentiert.<sup>40</sup>

### 2.2.1 Semantische Datenmodellierung

Nach dem Abschluss der Anforderungsanalyse kann mit der semantischen Datenmodellierung begonnen werden. Dabei werden alle konzeptuellen Anforderungen in die Überlegungen mit eingeschlossen. Durch die Systemunabhängigkeit während dieser Modellierungsphase, kann das Konzept frei von konkreten Anforderungen der Datenbanksysteme erstellt werden. Das semantische Datenmodell zeichnet sich durch einen hohen Abstraktionsgrad aus. Während dieser Modellierungsphase muss auf die Modellqualität sehr großen Wert gelegt werden, um aufwendige Anpassungsarbeiten von Beginn möglichst gering zu halten. Da innerhalb dieser Modellierungsebene die Anwendung aus der Sicht des Anwenders dargestellt wird, muss dessen Terminologie verwendet werden.

Für die semantische Datenmodellierung wird häufig das von Chen im Jahre 1976 entwickelte Entity-Relationship-Modell<sup>41</sup> (ER/M) angewendet.<sup>42</sup> Dieses Modell wurde bis heute immer wieder modifiziert, ist aber in seiner Grundstruktur erhalten geblieben und ist heute als „Standardmodellierungssprache“ im Datenbankdesign zu bezeichnen. Das ER/M besteht aus **Entitäten** (engl. entities), aus Objekten der realen Welt, **Attributen**, Eigenschaften der Entitäten, und **Beziehungen** (engl. relations) zwischen Entitäten.<sup>43</sup> So wurde beispielsweise jedes Buch von mindestens einem Autor verfasst. Solche Tatsachen werden im ER/M durch die Beziehung dargestellt (siehe Abbildung 4).

---

<sup>38</sup> Wedekind (2001), S. 144

<sup>39</sup> Totok (2000b), S. 190

<sup>40</sup> Lehmann (2001), S. 23

<sup>41</sup> weitere Ausführungen siehe Chen, Knöll (1991)

<sup>42</sup> vgl. Holthuis (2001), S. 142ff.

<sup>43</sup> vgl. Date (1986), S. 125ff.

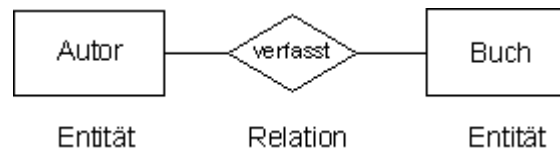


Abbildung 4: Entity-Relationship-Modell

Das Entity-Relationship-Modell (ERM), auf welches in Abschnitt 2.4 auf Eignung für multidimensionale Systeme untersucht wird, gehört zu der Gruppe der semantischen Datenmodelle.

### Datenqualität

Die Fragestellung der Datenqualität ist bereits bei der semantischen Datenmodellierung zu beachten und zu diskutieren. Bei der Extraktion aus den operativen Systemen muss sowohl eine gute Qualität als auch Quantität der Daten gewährleistet sein. Werden fehlerhafte oder gar unvollständige Daten in das Data Warehouse übertragen, kann keine qualitativ hochwertige Analyse der Daten erfolgen. Daraus ist abzuleiten, dass auf eine semantisch korrekte Modellierung zu achten ist. So sind unter dem Gesichtspunkt der multidimensionalen Abfrage oft Merkmalskombinationen möglich, die zwar semantisch darstellbar sind, aber in der realen Welt nicht vorkommen können. Dies kann z.B. der Fall sein, wenn ein Modell eines Produktes in einer bestimmten Region noch nicht eingeführt wurde, die Abfrage hierfür jedoch durchführbar ist. Dies liegt auch daran, dass manche Systeme keine Unterscheidung zwischen „0“ und „not available“ treffen. In diesem Fall wird also der Unterschied zwischen der semantischen und der mathematischen 0 nicht berücksichtigt.<sup>44</sup>

Grundsätzlich kann in multidimensionalen Informationssystemen zwischen fehlenden Daten unterschieden werden, die nicht erhoben worden sind und solchen, die semantisch nicht möglich sind.<sup>45</sup>

Der semantischen Datenmodellierung kommt die Aufgabe zu, eine verbesserte Diskussion zwischen der Informationsverarbeitung und den Fachabteilungen zu ermöglichen. Zur Kommunikation zwischen Entwickler und Anwender ist, über ein einheitliches Vokabular hinaus, eine identische Assoziation mit den verwendeten Begrifflichkeiten erforderlich.<sup>46</sup> Ein typisches Merkmal der semantischen Datenmodelle ist deshalb auch die grafische Repräsentationsform. In der Praxis hat es sich bewährt, einen Modellierungsworkshop durchzuführen. Dabei entwerfen die Data Warehouse-Entwickler zusammen mit den Anwendern das semantische Datenmodell. Auf der hohen Abstraktionsebene können alle Forderungen der Anwender dargestellt werden, und durch die Mitsprache der Anwender wird sowohl die Akzeptanz für das System, als auch Hintergrundwissen zum System-Entwurf vermittelt. Dadurch „fordern“ die Anwender nicht

<sup>44</sup> vgl. Totok (2000a), S. 237ff.

<sup>45</sup> Totok (2000a), S. 237

<sup>46</sup> Holthuis (2000), S. 184

nur, sondern sie sind in den Entwicklungsprozess aktiv eingebunden und können dabei im direkten Dialog mit den Entwicklern ein gemeinsames System entwerfen. Bei diesem Prozess sollte allerdings auf konstruktive Kritik und die Einhaltung der „Feedback-Regeln“ streng geachtet werden.

Ein semantisches Datenmodell vereinfacht die Entwicklung einer konsistenten und vollständigen Datenbasis im Data Warehouse und ist bei größeren und komplexeren Systemen unverzichtbar.<sup>47</sup>

Aus diesen Erkenntnissen schließt sich die Erkenntnis nach Totok an. Dieser schreibt: Ein semantisches Modell dient zur Begriffsklärung, zur Informationsbedarfsanalyse, zur Dokumentation und zur Datendefinition. Es kann als Diskussionsgrundlage zwischen Entwicklern und Mitarbeitern aus den Fachabteilungen fungieren. Semantische Modelle sind darüber hinaus (als Vorgabe für Data Dictionaries oder) als Navigationshilfe für die Endanwender nutzbar.<sup>48</sup>

Bei der semantischen Datenmodellierung muss in die Grundüberlegungen mit einfließen, wie viele und welche Dimensionen in dem Schema abgebildet werden müssen. Dabei ist auch die Anzahl der Hierarchiestufen zu definieren. Bei der Bestimmung der Dimensionen werden die Beziehungstypen festgelegt. Je nach Beziehungstyp können daraus weitere Modellierungsmaßnahmen abgeleitet werden. Eine 1:1-Beziehung ist bei der mehrdimensionalen Modellierung ungeeignet. Aus den 1:N-Beziehungen werden Dimensionshierarchien gebildet und M:N-Beziehungen werden schließlich durch zwei Dimensionen abgebildet.<sup>49</sup>

### **Zusammenfassung und Fazit**

Semantische Datenmodelle sind das Ergebnis des fachlichen Entwurfs (siehe Abbildung 2) und repräsentieren normierte Aussagen unabhängig von der Technologie, der später gewählten Datenspeicher und Endbenutzerwerkzeuge.<sup>50</sup>

Eine semantische Datenmodellierung bringt sowohl operative als auch dispositive und strategische Vorteile. Neben einer zentralen Koordination ist die Effizienzsteigerung bei der Systementwicklung sicher ein Hauptargument. Durch die verbesserte Kommunikation zwischen Anwender und Entwickler, auf Basis der erarbeiteten Modelle, kann eine bessere Einbindung der Fachbereiche und eine höhere Nutzerzufriedenheit realisiert werden. Die Modelle können im Entwicklungsprozess modifiziert und auch mehrfach verwendet werden, dies ist einer der Vorteile der systemunabhängigen Modellierung.<sup>51</sup>

In der Praxis zeigt sich aber immer wieder, dass eine völlig systemunabhängige Datenmodellierung nicht immer durchführbar ist, da Mängel der Tools anschließend nicht alle modellierbaren Systeme auch richtig umsetzen.<sup>52</sup>

---

<sup>47</sup> Holthuis (2000), S. 186

<sup>48</sup> Totok (2000a), S. 190

<sup>49</sup> vgl. Hahne (2004b)

<sup>50</sup> Lehmann (2001), S. 26

<sup>51</sup> vgl. Kaiser (1992), S. 208ff.

<sup>52</sup> vgl. Totok (2000a), S. 239

Semantische Datenmodelle reduzieren im allgemeinen die Entwicklungszeit. Ein semantisches Datenmodell dient vor allem der Vereinheitlichung der angewendeten Begrifflichkeiten zwischen den Anwendern und den Entwicklern des Data Warehouse. Als Fazit kann die Kostenreduzierung der Systementwicklung aufgrund der Summe dieser exemplarisch aufgeführten Vorteile genannt werden.

### **Ziele semantischer Datenmodellierung<sup>53</sup>**

- Abbildung der Datenbedeutung
- Strategische Informationsversorgung
- Implementierungsunabhängigkeit
- Funktionale Unabhängigkeit
- Kommunikation
- Transparenz
- Einheitliche Sprachregelung
- Qualitätssicherung

Daraus folgt:

Ziel der semantischen Datenmodellierung ist die Beschreibung von Daten eines relevanten Ausschnitts der realen Welt, um darauf aufbauend DV-technische Lösungen erzeugen beziehungsweise unterstützen zu können.<sup>54</sup>

### **2.2.2 Logische Datenmodellierung**

Für die logische Datenmodellierung multidimensionaler Strukturen wurde von Hahne ein Kriterienkatalog mit fünf Anforderungen aufgestellt.<sup>55</sup>

1. Strukturvielfalt  
bedeutet die Vielfalt der unterstützten Dimensionstypen mit den Komponenten der Hierarchie und der Attributierung.
2. Vielfalt der Operatoren  
beschreibt die Möglichkeit der Verknüpfung von Würfeln und der Bearbeitung von Würfeln.
3. Berechnungsmodelle  
sind die Grundlage für die Verdichtung innerhalb eines Würfels.
4. Temporale Aspekte  
haben die Strukturveränderungen im zeitlichen Ablauf eines Modells im Blick.
5. Normalformen und Integritätsbedingungen  
beschreiben Bedingungen an die Modelle, die entweder eine formale Begründung oder eine Begründung auf Basis des semantischen Modells haben.

---

<sup>53</sup> Kaiser (1992), S. 111ff.

<sup>54</sup> Holthuis (2000), S. 184

<sup>55</sup> Hahne (2002a), S. 59f.

Das logische Datenmodell muss bereits an die Speicherstruktur der zu verwendenden Datenbank angepasst werden. Das heißt, wie auch in Abbildung 2 dargestellt, dass zwischen dem semantischen und dem logischen Entwurf das Ziel-Datenbankmanagementsystem ausgewählt werden muss. Vor dieser Entscheidung muss der oben genannte Workshop mit der Fachabteilung abgeschlossen sein. Damit ist auch die Entwicklungsarbeit der Anwender abgeschlossen. Diese müssen von der technischen Modellierungsarbeit und der Implementierung unberührt bleiben. Die Anwender dürfen bei den anstehenden Tests des Systems wieder aktiv werden.

### **Die Hauptschritte im logischen mehrdimensionalen Datenbank-Entwurf<sup>56</sup>**

1. Identifizierung der Entitätstypen.
2. Identifizierung der Beziehungstypen.
3. Entwurf des Entity-Relationship-Diagramms mit Entitäten und Beziehungstypen.
4. Identifizieren der Wertetypen und Attribute.
5. Übertragen des Entity-Relationship-Diagramms in ein mehrdimensionales Datenmodell.
6. Entwurf der Datensatzformate.

Weitere Ausführungen siehe Chen, Knöll (1991).

### **Fazit:**

Der Erhalt der Informationen aus dem semantischen Datenmodell beim Übergang zum logischen Datenmodell ist sehr eng verbunden mit dem Kriterium der unterstützten Dimensionstypen in einem Würfel<sup>57</sup> und deren vielfältigen Attributen. Nicht nur die reine Unterstützung der verschiedenen Formen von Hierarchien in Dimensionen wird hier gefordert, sondern auch der Erhalt der rein semantischen Information der Hierarchie auf Ebene des logischen Modells. Wichtig ist es an einer gegebenen Struktur auf logischer Modellebene erkennen zu können, welche semantische Hierarchie zugrunde liegt.<sup>58</sup>

## **2.2.3 Ziele semantischer und logischer Datenmodellierung**

Die semantische und logische Datenmodellierungen können nicht vollkommen unabhängig voneinander betrachtet werden. Gewisse Überschneidungen und Abhängigkeiten infolge der Herleitung aus dem semantischen Modell ergeben somit eine gemeinsame Zieldefinition der Modelle.

Gemeinsame Ziele semantischer und logischer Datenmodellierung:<sup>59</sup>

- Operative Informationsversorgung
- Implementierungsgrundlage

---

<sup>56</sup> vgl. Chen, Knöll (1991), S. 59

<sup>57</sup> siehe Glossar

<sup>58</sup> Hahne (2002a), S. 43

<sup>59</sup> Kaiser (1992), S. 110f.



- Redundanzvermeidung
- Effizienzsteigerung
- Konsistenzkontrolle
- Datenunabhängigkeit
- Abbildungsfunktion

#### 2.2.4 Zusammenfassung und Fazit

In der Praxis wird immer wieder festgestellt, dass die Modellierung nicht völlig autark auf den drei Ebenen vorgenommen werden kann. In einem iterativen Modellierungsprozess wird zwar eine Ebene nach der anderen modelliert, aber für „Nacharbeiten“ muss auch auf die vorigen Modellierungsebenen zurück gewechselt werden können. Ebenso sind auch diverse Fähigkeiten bzw. Einschränkungen des einzusetzenden Datenbanksystems bereits auf der semantischen Modellierungsebene zu berücksichtigen.<sup>60</sup> Dies betrifft z.B. die Speicherstrukturen, aber auch das Speichervolumen.

Verschiedene Literaturquellen verwenden die Einteilung in semantische, logische und physische Ebene nicht einheitlich. Diverse Unterschiede sind hier immer wieder feststellbar. Es gibt keine allgemeingültige Definition der darzustellenden Inhalte auf den Ebenen, wie auch die unterschiedlichen Darstellungskonventionen verschiedene Darstellungsschwerpunkte setzen. Auch die Begrifflichkeit wird leicht unterschiedlich vorgenommen. So verwendet Hahne oder Totok die Untergliederung in semantisches, logisches und physisches Datenmodell.<sup>61</sup>

Synonym zu dem semantischen Datenmodell wird auch der Begriff des konzeptionellen oder konzeptualen Datenmodells verwendet. Das konzeptionelle Modell wird dabei manchmal auf das logische Modell übertragen. Eine einheitliche Gliederung wird bis zum gegenwärtigen Zeitpunkt noch nicht explizit vorgenommen.<sup>62</sup>

### 2.3 Von Fakten und Dimensionen über Fakt- und Dimensionsdaten zum Datenwürfel

Fakt- und Dimensionsdaten sind als die Grundlage in der multidimensionalen Datenbanktechnologie zu betrachten. Diese Datenarten werden im folgenden Text näher erläutert.

Faktdaten stellen die Kerndatenelemente mit meist quantitativer Natur dar, z.B. Umsatzzahlen oder Kostengrößen. Diese sind i.d.R. numerisch und stehen im Mittelpunkt der Datenanalyse. Oft werden die Fakten auch als betriebswirtschaftliche Variablen oder Kennzahlen (engl. variables, facts, measures) bezeichnet. Somit stehen – in Analogie zur multidimensionalen Matrix – in der Fakttable die gleichen Werte wie in den

---

<sup>60</sup> vgl. Lehner (2003), S. 54

<sup>61</sup> siehe Hahne (2002a), S. 10 oder Totok (2000a), S. 99f.

<sup>62</sup> vgl. Lehner (2003), S. 54

Zellen der Matrix. Die Anzahl der möglichen Datensätze in der Faktttabelle entspricht der Anzahl der belegten Zellen.<sup>63</sup>

**Fakten** (synonym: Basiskennzahlen) können von den **Kennzahlen** (synonym: abgeleitete Kennzahlen) abgeleitet werden.<sup>64</sup>

### Dimensionen

Eine Dimension ist innerhalb des multidimensionalen Datenmodells eine ausgewählte Entität, mit der eine Auswertungssicht eines Anwendungsbereichs definiert wird. Sie dient der eindeutigen, orthogonalen Strukturierung des Datenraums.<sup>65</sup> Eine Dimension ist also eine logische Gruppierung der Verdichtungsebene aus der betriebswirtschaftlichen Sichtweise der Anwender. Durch eine Dimension wird eine Achse des multidimensionalen Analyse- und Datenraums definiert. Die Summe aller Dimensionen wird auch als Dimensionschema bezeichnet.

Die kleinsten Informationseinheiten, die im Data Warehouse abgelegt werden, heißen atomare Daten. [...] Alle betrachteten Aufgliederungsrichtungen einer Kennzahl heißen Dimension. [...] Zusammenfassungen von atomaren Daten heißen Aggregationen. Ein System von elementaren Daten, Aggregationen und Aggregationen bereits aggregierter Daten ergibt eine Dimensionshierarchie.<sup>66</sup>

Ansätze, die eine multidimensionale ER-Modellierung verfolgen, unterscheiden in Anlehnung an die logische Modellierung mit Hilfe des Star-Schemas meist zwischen zentraler Faktrelation und mehreren Dimensionstabellen. Daher kann im ERM ein multidimensionales Modell als Verknüpfung eines zentralen Beziehungstyps mit einer Entity-Menge von Dimensionen interpretiert werden.<sup>67</sup>

Bei dem multidimensionalen Entity-Relationship Modell (ME/R Modell) handelt es sich um eine Modellierungsnotation, welche am bayrischen Forschungszentrum für wissensbasierte Systeme (FORWISS) entwickelt wurde. Bei dieser Notation wurden zur semantischen Modellierung die Elemente **Faktenrelation**, **Dimensionsebene** und **hierarchische Beziehung** hinzugefügt.<sup>68</sup>

### Würfel

Ein Würfel besteht aus Datenzellen, die eine oder mehrere *Kenngrößen* beinhalten. Die Datenzelle ist dabei Schnittpunkt der Dimensionen, die den Würfel aufspannen (Achsen des Würfels).<sup>69</sup>

---

<sup>63</sup> Behme, Holthuis, Mucksch (2000), S. 224

<sup>64</sup> vgl. Lehner (2003), S. 67f.

<sup>65</sup> Bauer, Günzel (2001), S. 172

<sup>66</sup> Hahne (1999), S. 149

<sup>67</sup> Totok (2000b), S. 191

<sup>68</sup> vgl. Totok (2000b), S. 191f. oder vgl. Totok (2000a), S. 125ff.

<sup>69</sup> Bauer, Günzel (2001), S. 174

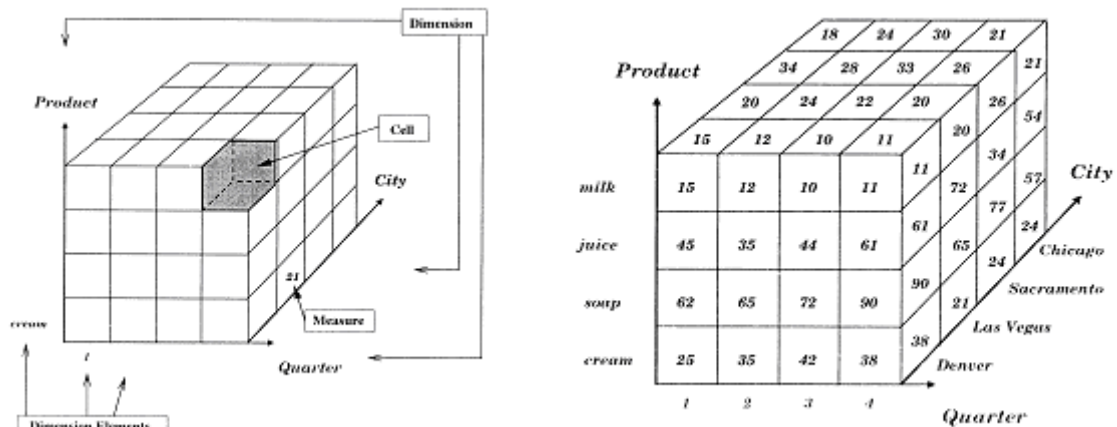
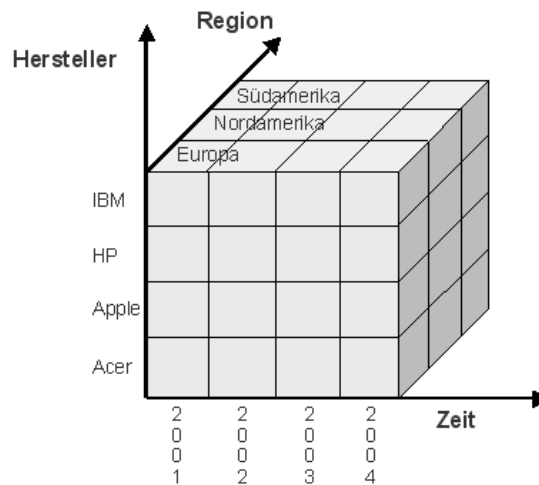
Abbildung 5: Multidimensionaler Datenwürfel mit Bestandteilen und Beispieldaten<sup>70</sup>

Abbildung 6: Mehrdimensionaler Datenwürfel oder Data-Cube der Fallstudie

Das Interessante an einem Data Cube ist, dass ein solcher Würfel vielfältig manipulierbar ist und sich aus ihm eine Reihe weiterer Aggregationen leicht berechnen lassen:<sup>71</sup>

1. Eine Projektion auf die xy-Ebene (Zeit X Hersteller) liefert die Verkäufe, jeweils über alle Regionen summiert; dies ist in Abbildung 6 dargestellt.
2. Eine Projektion auf die xz-Ebene (Zeit X Region) liefert die Verkäufe, jeweils über alle Hersteller summiert;
3. Eine Projektion auf die yz-Ebene (Hersteller X Region) liefert die Verkäufe, jeweils über alle Jahre summiert;
4. Eine Projektion auf die x-Achse (Zeit) liefert die Verkäufe, summiert über Hersteller und Regionen;

<sup>70</sup> Vossen (1999), S. 680f.

<sup>71</sup> in Anlehnung an Vossen (1999), S. 681f.

5. Eine Projektion auf die y-Achse (Hersteller) liefert die Umsätze, summiert über Zeit und Region;
6. Eine Projektion auf die z-Achse (Region) liefert die Verkäufe, summiert über Zeit und Hersteller;
7. Die Gesamtverkaufszahl über alle Jahre, Hersteller und Regionen ergibt sich durch Kollabieren des Würfels in die dem Ursprung am nächsten liegende Zelle, d.h. durch Summieren aller drei Dimensionen.

Aus relationaler Sicht kann man einen Würfel auch als eine um die Semantik der Dimensionen erweiterte Relation betrachten. Diese Erweiterung darf jedoch nicht unterschätzt werden, denn die implizite Verknüpfung des Würfels mit den Klassifikationshierarchien über die Granularität erlaubt nicht nur eine einfache und präzise Spezifikation von Operationen auf Würfeln, sie ermöglicht auch eine intuitive Abbildung multidimensionaler auf relationale Schemata sowie spezielle semantische Optimierungsverfahren.<sup>72</sup>

Zweidimensionale Würfel können grundsätzlich als Tabellen dargestellt werden. Auch ein dreidimensionaler Cube, wie in Abbildung 6, ist durch das menschliche Vorstellungsvermögen noch repräsentierbar. Bei höherdimensionalen Gebilden muss dieses auf eine für den Menschen darstellbare Form reduziert werden. Dies ist durch eine Reduktion auf ein zweidimensionales Gebilde gegeben.<sup>73</sup>

Spannt man aus den identifizierten Dimensionen ein räumliches Gebilde auf und weist jedem Schnittpunkt der Dimensionspositionen mit denen anderer Dimensionen eine Zelle zu, so entsteht eine multidimensionale Matrix, die auch Kreuztabelle genannt wird.<sup>74</sup> Dabei wird dann auch von einem Hypercube gesprochen. Die Struktur und Form dieses Hypercubes ergibt sich aus der Anzahl der Dimensionen, welche sowohl die Werte, welche die Faktdaten beinhalten, als auch die Charakterisierung der Matrix darstellen.<sup>75</sup>

Aus den Kennzahlen und dem Dimensionsschema kann somit der **multidimensionale Datenwürfel** als Modell abgeleitet werden. Dabei dürfen keine Abhängigkeiten zwischen den Attributen der Dimensionen auftreten.<sup>76</sup>

Aufbauend auf Fakten werden Kennzahlen definiert, die durch arithmetische Operationen aus den Fakten gewonnen werden. Bereits hier zeigt sich wieder deutlich der Unterschied zum klassischen Modellierungsansatz von Datenbanken, da Kennzahlen streng genommen als Sicht auf externer Ebene im Drei-Schema-Schichtenansatz be-

---

<sup>72</sup> Bauer, Günzel (2001), S. 175

<sup>73</sup> vgl. Hahne (2002a), S. 14

<sup>74</sup> Holthuis (2000), S.164

<sup>75</sup> vgl. Holthuis (2000), S. 164

<sup>76</sup> vgl. Lehner (2003), S. 67ff.

trachtet werden müssten und eigentlich nicht in den Entwurf des konzeptionellen Schemas Eingang finden dürften.<sup>77</sup>

## 2.4 Vom Entity-Relationship-Modell zum mehrdimensionalen Datenmodell – Ein Vorschlag zur Modellerstellung

Die Entity-Relationship-Modellierung in der dritten Normalform (3NF) unterscheidet sich in einigen Punkten von der mehrdimensionalen Datenmodellierung. Während beim DB-Design in der dritten NF keine Redundanzen vorkommen, werden diese bei der mehrdimensionalen DB-Entwicklung zur Performanceverbesserung ganz gezielt verwendet. Trotzdem sind viele Segmente relativ ähnlich.

Da relationale Datenbanken heute mit als „DER“ Standard in der Datenbanktechnik bezeichnet werden kann, und die mehrdimensionale Datenbank auf dieser Basis beruht, werden in vielen Unternehmen das vorhandene Know-How auf die analytischen Informationssysteme übertragen und somit die bewährte Technik beibehalten.<sup>78</sup>

Der identifizierende Schlüssel (Fremdschlüssel) der Faktentabelle setzt sich aus allen Schlüsseln der Dimensionstabellen zusammen. Die Schlüsselattribute der Dimensionstabelle referenzieren auf die Faktentabelle. Für diese Schlüsselkandidaten wird normalerweise ein künstlicher Schlüssel gebildet, d. h. eine Identifikationsnummer der einzelnen Dimensionstabellen geschaffen.

Fremdschlüsselbeziehungen zu benachbarten Tabellen (Dimensionen) dienen hierbei als eine Art Filter, um Ergebnismengen bei Data Warehouse-Abfragen auf die Faktentabelle nach bestimmten Kriterien einzuschränken.<sup>79</sup>

Alle in einer Faktentabelle repräsentierten Fakten hängen in einem Star-Schema voll funktional von der Menge der Primärschlüsselattribute der Dimensionstabellen ab.<sup>80</sup>

### Entity-Relationship-Modell

Das in der Folge dargestellte Entity-Relationship-Modell (ERM) basiert auf den Modellierungsüberlegungen nach Chen aus dem Jahr 1976.

**Entity-Relationship-Darstellung** (engl.: entity relationship representation): Zum Entwurf einer Datenbank ist es sehr hilfreich, eine informelle Problembeschreibung zunächst in ein **Entity-Relationship-Modell (ERM)** überzuführen, um daraus anschließend eine *logische Datenbankbeschreibung* zu entwickeln. Ein solches Modell besteht aus einer Beschreibung von Entitäten (engl. entities), Beziehungen (engl. relationships) und Attributen.<sup>81</sup>

---

<sup>77</sup> Lehner (2003), S. 68

<sup>78</sup> vgl. Hahne (2002a), S. 61

<sup>79</sup> Hahne (2002a), S. 84

<sup>80</sup> Lehner (2003), S. 87

<sup>81</sup> Schreier (2001), S. 184

**Entität:** Ein Exemplar aus der Menge der Sachverhalte, deren Semantik durch einen Entitäts-Typ klassifiziert werden.<sup>82</sup>

**Beziehung (Relation):** Ein Exemplar aus der Menge der Zusammenhänge, Verknüpfungen bzw. Wechselwirkungen zwischen Entitäten, deren Semantik ein Beziehungstyp beschreibt.<sup>83</sup>

**Attribut:** Mit einem Oberbegriff benannte Eigenschaftsart einer Menge von Informationen, denen die gleiche Semantik zugrunde liegt.<sup>84</sup>

Jede Entität hat bestimmte Merkmale (Attribute). Diese zur Beschreibung wichtigen Attribute sind z.B. für eine Entität **Produkt**:

- Produkt-ID (als Schlüsselmerkmal)
- Artikel
- Produktgruppe
- Produktfamilie
- Produktkategorie
- Bezeichnung
- Marke

Alle Entitäten und Beziehungen der realen Welt sind durch Relationen darzustellen. Eine vollständige Darstellung ist im Rahmen dieser Arbeit nicht möglich.<sup>85</sup> Die Entity-Relationship-Darstellung ist eine auch für DV-Laien relativ leicht verständliche Darstellung. Nicht zuletzt aus diesem Grund handelt es sich bei dem Modell um eine der verbreitetsten Modellierungssprachen zum DB-Entwurf.

### Das multidimensionale Entity/Relationship-Modell

In der Abbildung 7 wird ein multidimensionales E/R-Modell (mE/R-Modell) der Fallstudie dargestellt. Die zentrale Faktenrelation, hier in Form der Würfel Vertrieb in Verbindung mit Marketing-Kampagnen dargestellt, beinhalten die Faktdaten und die Kenngrößen. Über die Beziehungen wird bei dem Modell-Entwurf auf die Dimensionstabellen referenziert. Diese besitzen oft Variablen, welche durch Ellipsen darstellbar sind, wodurch die Dimension bei Bedarf näher spezifiziert werden kann. Die entscheidende Fragestellung bei der Informationsbedarfsanalyse zur Modellerstellung muss lauten:

“Was **braucht** der Anwender?“

In diesem Fall stellen sich also die Fragen:

1. Kosten je Kampagne in EUR?
2. Deckungsbeitrag in EUR?

---

<sup>82</sup> Kaiser (1992), S. 11

<sup>83</sup> Kaiser (1992), S. 14

<sup>84</sup> Kaiser (1992), S. 9

<sup>85</sup> weitere Ausführungen siehe Meier (1998)

3. Umsatz IST in Landeswährung (da die Kampagne weltweit durchgeführt werden kann)?
4. Produktgesamtkosten in EUR?

Um diese Fragen überhaupt beantworten zu können sind einige Vorbedingungen aus den Vertriebsdaten zu gewinnen.

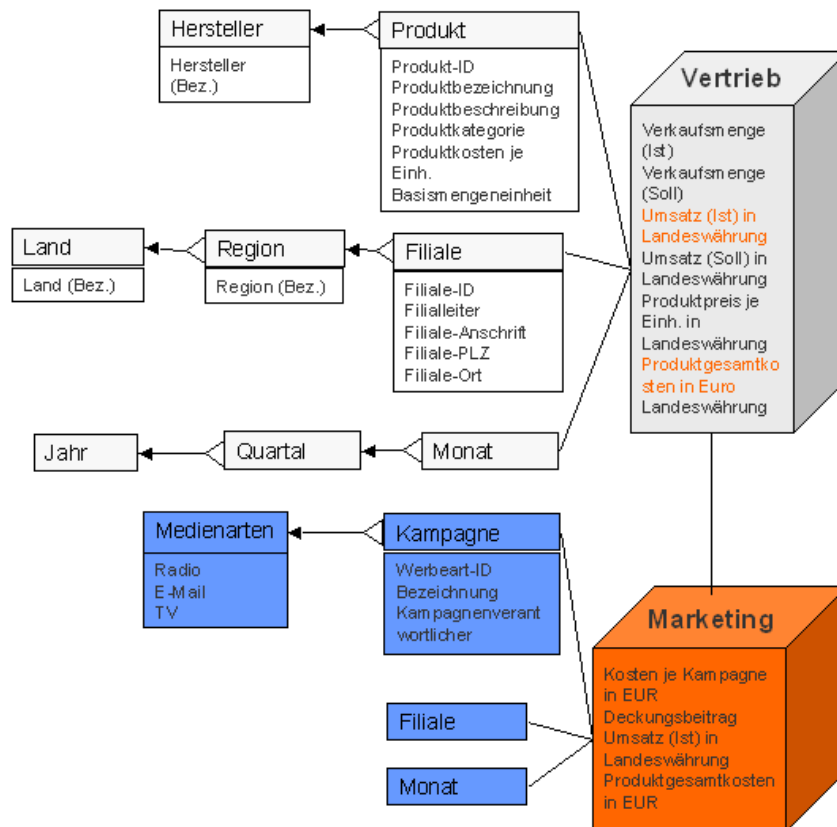


Abbildung 7: Multidimensionales Schema der Fallstudie in mE/R-Notation

**Dimensionshierarchien:** Eine Dimensionshierarchie besteht i.d.R. aus einer Menge an Dimensionen ( $D^1$ ;  $D^2$ ; ...,  $D^m$ ). Betrachtet man aus Abbildung 7 beispielsweise die Dimension >>Geographie<<, so besitzt diese Dimension drei Dimensionshierarchien. Diese entsprechen:  $D^1$  = Filiale;  $D^2$  = Region;  $D^3$  = Land.

**Kennzahlen:** Die Kennzahlen bestehen i.d.R. aus einer Menge an Kennzahlen ( $M^1$ ;  $M^2$ ; ...,  $M^n$ ). Betrachtet man aus Abbildung 7 beispielsweise die Faktentabelle >>Vertrieb<<, so beinhaltet diese eine variable Menge an Kennzahlen.

Die *Dimensionshierarchien* und die *Kennzahlen* spannen gemeinsam einen Datenwürfel auf.

Dabei darf allerdings keine „Orthogonalität“ innerhalb der Dimensionen auftreten. Dies bedeutet, dass keine funktionalen Abhängigkeiten zwischen den Attributen der unterschiedlichen Dimensionen auftreten dürfen.<sup>86</sup>

<sup>86</sup> vgl. Lehner (2003), S. 64ff.

Die Notation des mE/R-Modells stellt eine Erweiterung des bekannten ER-Ansatzes dar.<sup>87</sup> Die mE/R-Notation wurde als spezielle Modellierungstechnik für multidimensionale Schemata entwickelt.<sup>88</sup>

Die Grundidee des mE/R-Ansatzes ist dabei wie folgt: Um eine naturgemäße Darstellung der multidimensionalen Semantik zu erlauben, wird das E/R-Modell entsprechend spezialisiert und geringfügig erweitert (evolutionärer Ansatz). Dazu gibt es generell mehrere Möglichkeiten. Die spezielle Lösung für die mE/R-Notation wurde dabei von folgenden Überlegungen beeinflusst:<sup>89</sup>

- *Spezialisierung des E/R-Modells:* Alle eingeführten Elemente der mE/R-Notation sollten Spezialfälle der ursprünglichen E/R-Konstrukte sein. Dadurch wird weder die Flexibilität noch die Ausdrucksmächtigkeit des E/R-Modells reduziert.
- *Minimale Erweiterung des E/R-Modells:* Das spezialisierte Modell sollte für einen erfahrenen E/R-Modellierer leicht zu erlernen und zu benutzen sein. Folglich sollte die Zahl der zusätzlichen Elemente so gering wie möglich sein. Eine minimale Menge von Erweiterungen gewährleistet ferner den Transfer vorhandener wissenschaftlicher Ergebnisse (z.B. formale Fundierung) vom E/R-Modell auf den Fall des mE/R-Modells, weil nur die mE/R-spezifischen Erweiterungen betrachtet werden müssen.
- *Darstellung der multidimensionalen Semantik:* Trotz der Forderung nach Minimalität sollte die Spezialisierung mächtig genug sein, um die grundlegende multidimensionale Semantik ausdrücken zu können. Damit sind im Wesentlichen die Unterscheidung zwischen Klassifikationsschema und Würfelstruktur (also qualifizierenden und quantifizierenden Daten) und die hierarchische Struktur der Klassifikationen gemeint.

Das multidimensionale Datenmodell stellt einen Kompromiss zwischen Einfachheit und Verständlichkeit auf der einen Seite und einer umfassenden Modellierungsmächtigkeit auf der anderen Seite dar. Die Einfachheit des Modells erlaubt es dem Anwender, mit Standardwerkzeugen auf den Datenbestand zuzugreifen und Anfragen ohne Fachkenntnisse zu formulieren. Dies führt allerdings dazu, dass sich komplexere Zusammenhänge der Anwendungsdomäne nicht immer adäquat im multidimensionalen Modell abbilden lassen.<sup>90</sup>

Moody und Kortink erläutern, dass es sich bei dem mehrdimensionalen Modell um ein eingeschränktes E/R-Modell handelt. Nach dieser Auffassung wird ein E/R-Modell in der Teilmenge denormalisiert als mehrdimensionales Modell bezeichnet.

---

<sup>87</sup> vgl. Bauer, Günzel (2001), S. 157

<sup>88</sup> Bauer, Günzel (2001), S. 160

<sup>89</sup> Bauer, Günzel (2001), S. 160 f.

<sup>90</sup> Bauer, Günzel (2001), S. 182



### **Mehrdimensionale Modellierung**

Die mehrdimensionale Datenmodellierung ist eine speziell für die Data Warehouse Entwicklung entstandene Datenbank-Design Methode.

Als Vorteile der multidimensionalen Modellierung gilt:<sup>91</sup>

- Es werden Datenbankstrukturen gebildet, welche für den Anwender einfach zu nutzen und zu verstehen sind, und von den Nutzern abgefragt werden können.
- Optimierte Abfrage-Performance (nicht unbedingt bei der Update-Performance).

Die mehrdimensionale Modellierung wurde in der Vergangenheit die dominierende Praxismethode um Data Warehouses zu modellieren. Dieser Erfolg wird durch die Einfachheit und dadurch guten Verständlichkeit für den Anwender zurückgeführt, da dieser das mehrdimensionale Modell direkt lesen und übertragen kann. Die deutliche Reduzierung der Modellkomplexität des mehrdimensionalen Modells im Vergleich zu anderen Datenbankmodellen ist ein großer Vorteil dieser Modellfamilie.

Ausgehend vom E/R-Modell wird das mehrdimensionale Modell in vier Schritten erstellt:

Schritt 1: Entity definieren

Schritt 2: Spezifiziertes Star-Schema erstellen

Schritt 3: Detaillierte Fakten-Tabelle erstellen

Schritt 4: Detaillierte Dimensionstabelle erstellen

Durch die Transformation vom E/R-Modell in ein mehrdimensionales Modell wird eine Verbindung zwischen OLAP und OLTP gebildet. Dies ist für den Aufbau eines Data Warehouse und für das bessere Verständnis durch die Anwender und Entwickler von großer Bedeutung. Ein wichtiges Kriterium ist hierbei, die „richtigen“ Dimensionen für den DW-Aufbau zu wählen.<sup>92</sup>

### **Zusammenfassung**

Als Nachteil eines multidimensionalen Datenmodells im Vergleich zu dem Entity-Relationship-Modell stellt sich die geringere Flexibilität im direkten Vergleich der Modelle heraus. Allerdings ist die leichte Navigation durch den gesamten Datenbestand mit dem multidimensionalen Modell einfacher, was gerade bei komplexen Strukturen einen erheblichen Vorteil darstellt.<sup>93</sup>

Der Hauptunterschied zwischen dem multidimensionalen und dem relationalen Modell ist die zusätzliche Semantik, durch die die Beziehungen zwischen den Klassifikationsstufen einer Dimension untereinander, zwischen den Würfeln und den Klassifikationsstufen seiner Dimensionen sowie zwischen verschiedenen Würfeln zum Bestandteil des Modells gemacht werden.<sup>94</sup>

---

<sup>91</sup> vgl. Moody, Kortink (2003), S. 8

<sup>92</sup> vgl. Moody, Kortink (2003), S. 8ff.

<sup>93</sup> vgl. Lehner (2003), S. 64

<sup>94</sup> Bauer, Günzel (2001), S. 194

### 3 Multidimensionale Strukturen für relationale Datenbanken

Bei den Managementunterstützungssystemen erfolgen Abfragen und Analysen nahezu immer über mehr als zwei Dimensionen. Daher kommt die Bezeichnung der Multidimensionalität.<sup>95</sup> Dies wird anhand eines Beispiels der Fallstudie verdeutlicht: Ein Abteilungsleiter möchte die Umsätze des Produkts Notebook des Herstellers HP für die Region Nordamerika im Jahr 2004 wissen. Für eine solche relativ einfache Abfrage reicht die Speicherung in einer Tabelle nicht mehr aus. Man greift deshalb auf die multidimensionale Datenhaltung, welche in den folgenden Abschnitten näher erläutert wird, zu. Als eine Visualisierungsmöglichkeit wird hierzu der dreidimensionale Würfel (oder Cube) (Abbildung 8) als intuitives Modell verwendet. Die Multidimensionale Datenanalyse ist die fundamentale Grundidee eines Data Warehouses. Mehrdimensionale Abfragen bilden dabei das Grundkonzept, welches zum besseren Verständnis durch den Datenwürfel visualisiert werden kann. Die Kanten des Würfels sind in der Praxis selten genau gleich lang, aber zur Visualisierung hat sich das Bild des Würfels etabliert.

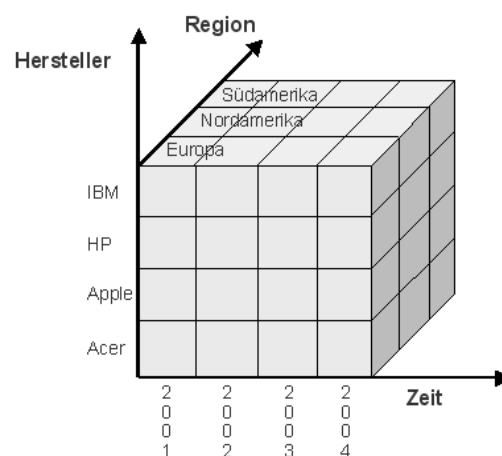


Abbildung 8: Dreidimensionaler Daten-Würfel aus der Fallstudie

Unter Multidimensionalität versteht man „die Art der logischen Anordnung quantitativer Größen bzw. betriebswirtschaftlicher Variablen (wie z.B. Umsatz- oder Kostengrößen), die durch mehrere sachliche Kriterien (wie z.B. Perioden, Kunden, Artikel, Niederlassungen oder Regionen) beschrieben sind. Versinnbildlicht erscheinen diese Größen dann als Sammlung von Würfeln, wobei die einzelnen Dimensionen durch die zugehörigen textindizierten Würfelkanten repräsentiert werden“.<sup>96</sup>

<sup>95</sup> vgl. Holthuis (2000), S. 150f.

<sup>96</sup> Gabriel, Röhrs (2003), S. 337

Die Definition eines **multidimensionalen Datenmodells** lautet: System von Strukturen und Operatoren zur Modellierung von Dimensionen und Klassifikationshierarchien innerhalb eines Analysekontextes.<sup>97</sup>

Ziel der multidimensionalen Abbildung ist es, eine Möglichkeit zu finden, multidimensionale Strukturen in einem relationalen Datenmodell abzubilden, so dass:<sup>98</sup>

- möglichst wenig anwendungsbezogene Semantik, die im multidimensionalen Modell enthalten ist, bei der Abbildung verloren geht (z.B. Klassifikationshierarchien),
- die Übersetzung von multidimensionalen Anfragen effizient geschehen kann,
- die Abarbeitung der übersetzten Anfragen durch das relationale Datenbanksystem auf möglichst effiziente Weise erfolgt und
- die Wartung (z.B. das Laden neuer Daten) der entstandenen Tabellen einfach und schnell erledigt werden kann.

Die vorgestellte Darstellungsform des Datenwürfels muss in eine für das Datenbank-System speicherbare Form übertragen werden.

*A row in a fact table corresponds to a measurement. A measurement is a row in a fact table. All the measurements in a fact table must be at the same grain.*<sup>99</sup>

Jede Relation kann als multidimensionaler Würfel betrachtet werden. Für diese Sichtweise werden die Spalten der Relation als Dimensionen des Würfels aufgefasst.<sup>100</sup> Jede Dimension wird also in eine Spalte der Tabelle übertragen. Dabei wird die Zeit in diesem Beispiel in Jahre erfasst. Diese Sichtweise und der Zusammenhang zwischen Würfel und Tabelle wird in der folgenden Abbildung verdeutlicht.

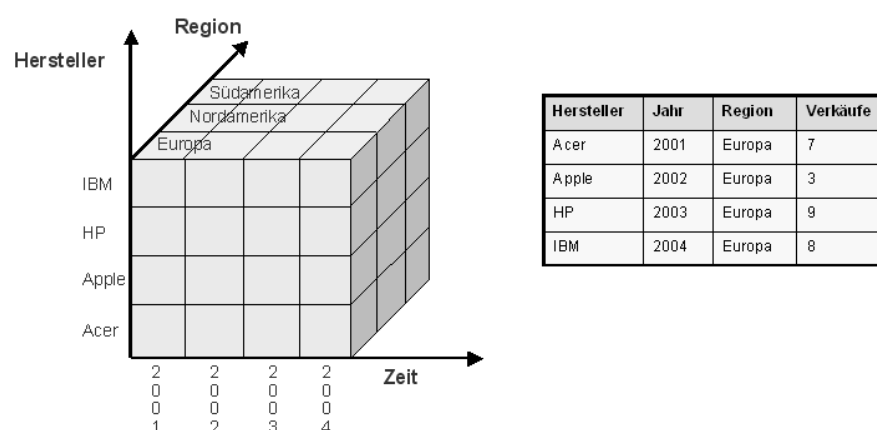


Abbildung 9: Dualismus von Würfel und Tabelle<sup>101</sup>

<sup>97</sup> Bauer, Günzel (2001), S. 519

<sup>98</sup> Bauer, Günzel (2001), S. 198

<sup>99</sup> Kimball (1996), S. 17

<sup>100</sup> vgl. Bauer, Günzel (2001), S. 198ff.

<sup>101</sup> in Anlehnung an Bauer, Günzel (2001), S. 199

Das „*Übersetzungs-Problem*“ liegt in der Tatsache begründet, dass die Multidimensionalität der Abfrage auf eine meist relationale, also zweidimensionale Struktur, zu transformieren ist. Unter diesem Aspekt ist auch die Abfrageperformance zu beachten. Bei der Systemerstellung ist auf eine homogene Arbeit aller Teilsysteme zu achten, damit sich ein eventuell schlecht designtes Teilsystem nicht all zu nachteilig auf das Gesamtsystem auswirkt.

Somit sind an **multidimensionale Datenbanksysteme** besondere Anforderungen gebunden. Multidimensionale Datenbanken müssen alle Anforderungen jeder Datenbank erfüllen (Mehrbenutzerfähigkeit, Konsistenz der Transaktion, Daten-Sicherheit und -Richtigkeit, etc.). Außerdem müssen multidimensionale Datenbanken aufgrund der Data Warehouse Anforderungen in die Schichtenarchitektur (Server-Client-Architektur) eingebunden werden. Näheres zu dieser Schichtenarchitektur kann in Bauer, Günzel (2001)<sup>102</sup> (Kapitel 4.1 bzw. 4.2) nachgelesen werden. Zusammengefasst können multidimensionale Datenbanksysteme nach folgenden Kriterien charakterisiert werden:<sup>103</sup>

- Offenheit
- Front-End-Tools für die Analyse der Daten
- Client-/ Server-Umgebung
- Fragestellung der Zeit (Standarddimensionen mit vordefinierter Struktur in der Datenbank-Software – im Sinne von Schablonen oder Templates)
- keine Standards der Funktionalitäten
- „DIE“ geeignete Struktur der MDBS gibt es nicht
- meist sind MOLAP Funktionalitäten integriert

### Denormalisierung

Relationale Datenbanken werden zur Vermeidung von Redundanzen und der Wahrung der referentiellen Integrität in Datenbanken normalerweise in der dritten Normalform verwaltet.<sup>104</sup> Um die Performance und das Antwortzeitverhalten möglichst zu verbessern wird im Datenbankdesign öfters eine Denormalisierung vorgenommen. Dies ist in Data Warehouse-Systemen eine gängige Design-Methode. Das heißt, eine gewollte Redundanz wird eingeführt, um einen möglichst schnellen Zugriff zu gewährleisten. Auf dieses Mittel wird auch bei dem Star-Schema-Modell (siehe Kapitel 3.1.1) zurückgegriffen.<sup>105</sup> Ein Merkmal einer Tabelle ist *redundant*, wenn einzelne Werte dieses Merkmals innerhalb der Tabelle *ohne Informationsverlust weggelassen* werden können.<sup>106</sup> In Tabelle 1 ist eine redundante Datenspeicherung in der Relationendarstellung abgebildet.

---

<sup>102</sup> weitere Ausführungen siehe Bauer, Günzel (2001), S. 117ff.

<sup>103</sup> vgl. Holthuis (2001), S. 186ff.

<sup>104</sup> vgl. Meier (1998), S. 34ff.

<sup>105</sup> vgl. Mucksch, Behme (2000), S. 42ff.

<sup>106</sup> Meier (1998), S. 34

Tabelle 1: Redundante Datenspeicherung in einer Tabelle<sup>107</sup>

Gesamtvertrieb	Region	Niederlassung	Filiale	Verkäufer
Deutschland	Nord	Hamburg	Kiel	18
Deutschland	Süd	München	Isartor	3
Deutschland	Süd	München	Isartor	2
Deutschland	Süd	Nürnberg	Breite Gasse	7

Die Modellierung der Speicherstrukturen in multidimensionalen Datenbanken erfolgen von den Entwicklern ganz bewusst in denormalisierter Form um eine gute Performance der Abfragen zu erzielen. Dadurch sind die Abfragen für den Anwender einfacher zu formulieren.

Nach Inmon treten Denormalisierungen im Bereich Data Warehouse in unterschiedlichen Formen und Auswirkungen auf das System auf:<sup>108</sup>

- Denormalisierung der 1:N-Beziehungen
- Redundanzen in der Mastertabelle (Faktentabelle)
- Selektive Redundanzen senken die Anzahl der Datenbankzugriffe
- Daten, welche nacheinander abgerufen werden sind in Datenblöcken gespeichert

Ralph Kimball nennt für die Modellierung mehrdimensionaler Datenbanken sogar fünf zu beachtende Mythen:<sup>109</sup>

1. Mehrdimensionale Modelle und Data Marts sind nur für Summendaten geeignet.
2. Mehrdimensionale Modelle und Data Marts sind Bereichsspezifisch, nicht Unternehmensweit.
3. Mehrdimensionale Modelle und Data Marts sind nicht skalierbar.
4. Mehrdimensionale Modelle und Data Marts sind nur adäquat, falls ein vorhersehbares Anwendungsmodell existiert.
5. Mehrdimensionale Modelle und Data Marts sind nicht integrierbar.

<sup>107</sup> in Anlehnung an Wieken (2002), S. 173

<sup>108</sup> vgl. Mucksch, Behme (2000), S. 43f.

<sup>109</sup> Kimball (1996), S. 24ff.

### Fakten- und Dimensionstabellen

Dimensionsdaten unterliegen durch normale Geschäftstätigkeiten einem stetigen Wandel. Im Gegensatz dazu bleiben Faktdaten in aller Regel konstant.

→ **Faktentabellen:** *Fact tables express the many-to-many relationships between dimensions in dimensional models.*<sup>110</sup>

→ **Dimensionstabellen:** *Dimension tables are the entry points into the fact table. Robust dimension attributes deliver robust analytic slicing and dicing capabilities. The dimensions implement the user interface to the data warehouse.*<sup>111</sup>

Tabelle 2: Eigenschaften der Fakten- und Dimensionstabellen

Eigenschaften Faktentabelle	Eigenschaften Dimensionstabelle
Eine Tabelle (im Star-Schema)	Mehrere Tabellen
Enthält die meisten Reihen eines Data Warehouse	Enthält gewöhnlich nur einen Bruchteil der gesamten Daten
Die Primärschlüssel aller Dimensionstabellen bilden den (zusammengesetzten) Primärschlüssel der Faktentabelle	Gewöhnlich bildet eine einzige Spalte den Primärschlüssel
Bewegungsdaten (Werte der Kennzahlen und Dimensionen)	Stammdaten und Bewegungssätze (z.B. Produktdaten)
Spalten sind numerisch und additiv	Spalten sind textuell und beschreiben die entsprechenden Dimensionen
Enthält historische Daten	Qualifizierende Informationen (beurteilende, einstufige Daten)
	Die Definition der Felder erfolgt in einer Faktentabelle

Die drei wesentlichen Funktionen der Dimensionstabellen:<sup>112</sup>

1. Sie beschreiben die Fakten, um daraus sinnvolle Aussagen entstehen zu lassen.
2. In ihnen sind die Suchkriterien festgelegt, nach denen die Fakten sinnvoll ausgewertet werden können.
3. Sie definieren die Hierarchien, entlang derer die Verdichtungsstufen für die Auswertungen festgelegt werden.

<sup>110</sup> Kimball (2002), S. 19

<sup>111</sup> Kimball (2002), S. 20

<sup>112</sup> Hahne (2002a), S. 83

Jede Dimensionstabelle besteht aus je einem ID-Schlüssel, welcher den Primärschlüssel für eine Beziehung zu der Faktentabelle darstellt. Die Attribute bestehen aus den Merkmalen der Dimensionen. Eine Dimensionstabelle im Beispiel Geographie, besteht also aus Geographie-ID als Schlüssel und Land, Region sowie Stadt als Attribute.<sup>113</sup> Vereinfacht ausgedrückt, werden die Daten der Dimensionstabellen dazu verwendet, die Bedeutung der Daten von Faktentabellen einzustufen, oder synonym zu qualifizieren.

Für die mehrdimensionale Datenmodellierung stehen verschiedene adäquate Modellierungstechniken zur Verfügung, welche eine strukturierte Betrachtung auf Würfel und Dimensionen mit Hilfe der Modelle erlaubt.

### 3.1 Modellierungsmethoden für relationale Datenbanken

Wie in Tabelle 3 dargestellt ist, kann beim Erstellen des konzeptionellen Entwurfs auf die Varianten der Entity-Relationship-Methode zurückgegriffen werden. Dabei ist in diesem Stadium noch unerheblich, ob ein klassischer relationaler oder multidimensionaler Datenbankentwurf erstellt werden soll. Um die Transformation von semantischer zu logischer Ebene einfach zu gestalten, ist ein durchgängiger multidimensionaler Entwurf zu bevorzugen, wird aber nicht zwingend vorausgesetzt.

Tabelle 3: Vergleich von relationalem und multidimensionalem Datenbankentwurf<sup>114</sup>

	<b>Klassisch relationaler Datenbankentwurf</b>	<b>Multidimensionaler Datenbankentwurf</b>	
Konzeptionelles Schema (semi-formal)	Varianten der Entity-Relationship-Methode	diverse Ansätze für Entwurfsnotationen, z.B.: mE/R, mUML, ADAPT, ...	
Logisches Schema (formal)	Relationen mit Attributen	Datenwürfel mit Summenattributen: Fakten und Kennzahlen	
		Dimensionshierarchien mit Kategorienattributen: klassifikatorische und beschreibende Attribute	
Internes Schema	Speicherorganisation (Primär- und Sekundärindexe, Partitionierung, ...)	Relationale Speicherorganisation (ROLAP): Schemamuster nach Star/Snowflake	Multidimensionale Speicherorganisation (MOLAP): Native Implementierung

<sup>113</sup> vgl. Schinzer, Bange, Mertens (1999), S. 48ff.

<sup>114</sup> Lehner (2003), S. 56

Unter dem Begriff „Star-Schema“ (Kapitel 3.1.1) werden verschiedene logische Datenmodelle zur Modellierung eines Data Warehouse auf der Basis relationaler Datenbanken subsumiert. Diese Modelle zielen darauf ab, mehrdimensionale Datenstrukturen in Relationenmodellen abzubilden.<sup>115</sup>

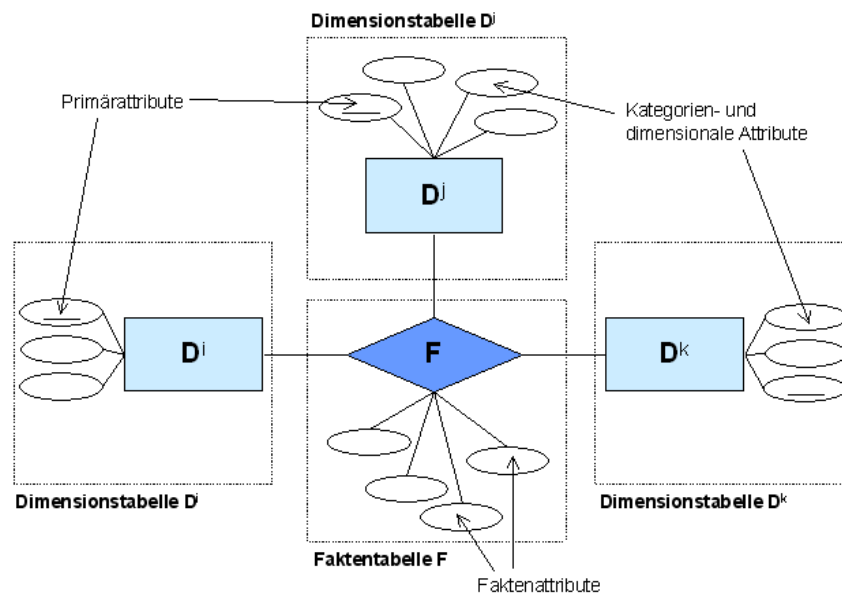


Abbildung 10: Übergang von ER-Diagramm zum Star-Schema<sup>116</sup>

### 3.1.1 Star-Schema

Bei dem Star-Schema handelt es sich um ein Modell für relationale Datenbanken. Die Zentrale Basis des Star-Schemas bildet die Faktentabelle. Diese besteht aus mehreren Dimensionsschlüsseln. Daraus resultiert, dass es genau so viele Dimensionstabellen im Star-Schema gibt, wie die Faktentabelle Dimensionsschlüssel aufweist. Die Dimensionsschlüssel sind in der Faktentabelle typischerweise jeweils zu anfangs angeordnet. Darunter folgen die Kennzahlen, mit welchen i.d.R. Berechnungen durchgeführt werden. Typische Vertreter von Kennzahlen sind z.B. Umsatz oder Absatzmenge. In der Faktentabelle wird jede Transaktion als ein Datensatz erfasst. Dies bedeutet, dass die Faktentabellen häufig relativ schnell ein großes Volumen annehmen. Jeder Datensatz der Faktentabelle besteht aus Schlüsselattributen und numerischen Attributen. Durch die sternförmige Anordnung der Dimensionstabellen um die Faktentabelle (vgl. Abbildung 11) hat dieses Schema seinen Namen Star-Schema erhalten.<sup>117</sup>

<sup>115</sup> Hahne (1999), S. 152

<sup>116</sup> Lehner (2003), S. 86

<sup>117</sup> vgl. Schinzer, Bange, Mertens (1999), S. 48ff.



Eigenschaften des Star-Schemas:<sup>118</sup>

- Die Faktentabelle ist in der dritten Normalform (3NF). Eine Übertretung in die zweite Normalform (2NF) würde sich bei Abfragen fehlerhaft auswirken. Doppelte Datenberücksichtigungen sind dann die Folge. Die Faktentabelle bildet eine Schnittstelle mit N-Beziehungen, dabei steht N für die Anzahl der Beziehungen, zu den Dimensionstabellen. In der Faktentabelle sind alle Schlüssel der einzelnen Dimensionstabellen enthalten.
- Es gibt eine oder mehrere Entitäten, welche als Dimensionstabellen gespeichert werden. Diese stehen alle in einer oder mehreren eins-zu-vielen Schlüsselbeziehung. Dimensionstabellen besitzen jeweils ein Schlüsselmerkmal und werden wenigstens in der zweiten Normalform (2NF) gespeichert. Transitive Abhängigkeiten (Verletzungen der 3NF) sind erlaubt, die Regeln der ersten und zweiten Normalform sind jedoch zwingend einzuhalten.

Die Speicherung von multidimensionalen Daten in einem Star-Schema bedeutet, dass es aufgrund der vielfältigen Verknüpfungsmöglichkeiten unter den Dimensionen zu einer enorm großen Anzahl von Datensätzen in den Fakttabellen kommen kann. Je größer die Datenbestände in einer Datenbank werden, desto negativer wirkt sich dies auf die Leistungsfähigkeit und das Antwortzeitverhalten des Systems aus.<sup>119</sup>

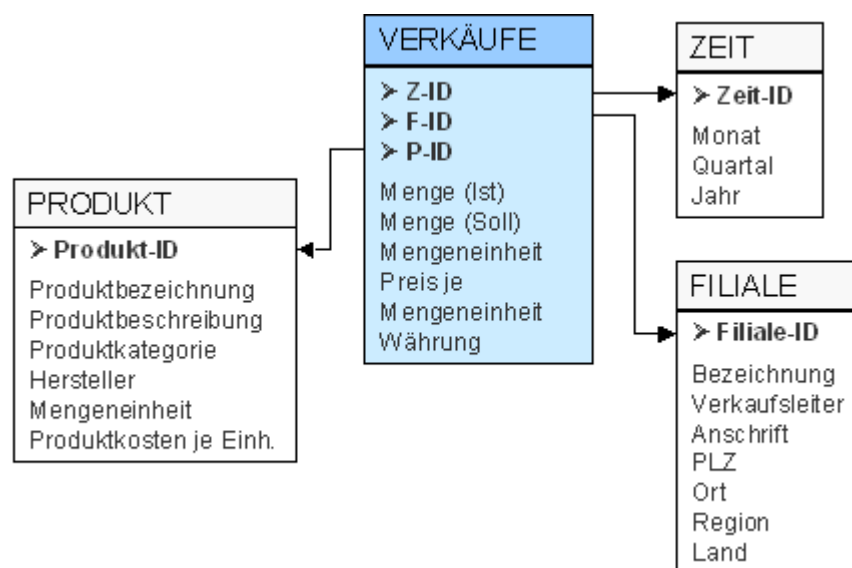


Abbildung 11: Beispiel eines Star-Schemas

Hahne schreibt über das Star-Schema: „Hierunter ist eine facettenreiche Vielzahl von Varianten einer relationalen Modellklasse zu verstehen, deren Ursprung in der mehr-

<sup>118</sup> vgl. Moody, Kortink (2003), S. 13

<sup>119</sup> Behme, Holthuis, Mucksch (2000), S. 229

dimensionalen Betrachtungsweise liegt. Die formale Basis für diese Modelle bildet das Relationenmodell, ...<sup>120</sup>

Das klassische Star-Schema entwickelt sich im Laufe der Zeit als ein Standard zur Modellierung mehrdimensionaler Strukturen in relationalen Datenbanken. Dabei werden die quantifizierenden Informationen (messbare Daten), man spricht dabei auch von Bewegungsdaten, in der Faktentabelle gespeichert. Bewertungsdaten sind Werte der Kennzahlen bzw. Variablen. Alle Kennzahlenwerte sind zunächst in genau einer großen Faktentabelle gespeichert. Das Schema der Faktentabelle muss die Beziehungen zu den Dimensionstabellen berücksichtigen. Die qualifizierenden Informationen (beurteilende, einstufende Daten) werden in den Dimensionstabellen als Dimensionseinträge gehalten. Bei diesen Daten handelt es sich um Stammdaten, die Bewegungsdatensätze (z.B. Produktdaten) beschreiben. Die Dimensionstabellen sind im Modell satellitenartig um die Faktentabelle angeordnet.<sup>121</sup>

Ein wesentliches Unterscheidungskriterium für Star Schema-Varianten ist der Grad der Normalisierung und der Schlüsselbildung der Dimensionstabellen, sowie die Modellierung der Dimensionsstrukturen der jeweiligen Dimensionen z.B. in Form von rekursiven Beziehungen oder durch festgelegte Gruppierungsattribute.<sup>122</sup>

Jedes Attribut des primären Gesamtschlüssels der Faktentabelle ist gleichzeitig der Fremdschlüssel einer Dimensionstabelle.<sup>123</sup> Verbunden sind die Fakttabellen und die Dimensionstabellen somit über gemeinsame Schlüsselattribute. Die Besonderheit dieser Datenstruktur liegt in der redundanten Datenspeicherung innerhalb der Dimensionstabellen.

### **Merkmale des Star-Schemas**

Star-Schemata besitzen die folgenden Eigenschaften:

- Jede Faktentabelle enthält Attribute
- Die Faktentabelle enthält wenige „not available“ Felder (vollbesetzten Matrix)
- Die Faktentabelle integriert N:M-Beziehungen, dadurch ergeben sich verschiedene Redundanzen
- Jede Dimensionstabelle steht in einer 1:N-Beziehung zu der Faktentabelle
- Die 1:M-Beziehung wird über einen Schlüssel der Dimensionstabelle und einen Fremdschlüssel der Fakttable hergestellt
- Dimensionstabellen sind oft nur dünn besetzt („not available“-Felder) (dünn besiedelte Matrix)
- Dimensionstabellen benötigen weniger Speicherplatz als Faktentabellen

---

<sup>120</sup> Hahne (2002a), S. 81

<sup>121</sup> vgl. Hahne (2002a), S. 82f.

<sup>122</sup> Hahne (2002b), S. 414

<sup>123</sup> Totok (2000a), S. 174

**Vorteile des Star-Schemas<sup>124</sup>**

- Durch die denormalisierte Speicherform sind bei Abfragen deutlich weniger Verknüpfungsoperationen (join-Operationen) nötig. Dadurch entsteht eine kürzere Zugriffszeit.
- Reduzierung der fehlerhaften Benutzeranfragen, da eine vereinfachte Sicht auf das Datenmodell vorliegt.
- Multidimensionale Datenstrukturen können sehr leicht beschrieben werden.
- Integration in bereits bestehende relationale Informationssysteme ist realisierbar.

*The fixed structure of the star schema limits the queries that can be written to the dimensions that have been defined. This means the designer must have a good idea in advance about the sort of questions users will want to ask.*<sup>125</sup>

Das Star-Schema ist ein Ansatz zur Speicherung multidimensionaler Datenstrukturen in relationalen Datenbanksystemen (ROLAP). Die grundlegende Prämisse des Star-Schemas ist die Klassifikation von Daten in zwei Gruppen: *Fakt- und Dimensionsdaten*. [...] Im Zentrum des Star-Schemas steht die Fakttable mit den entsprechenden Daten. Um diese Fakttable herum ist für jede repräsentierte Dimension eine Tabelle angeordnet. Verknüpfungen bestehen nur mit der Fakttable; die Dimensionstabellen sind nicht miteinander verknüpft. Hieraus entsteht eine sternförmige Anordnung der Tabellen mit der Fakttable als Zentrum des Sterns und den Dimensionstabellen als Endpunkte der Zacken.<sup>126</sup>

**Zusammenfassung**

Durch das zugrundeliegende relationale Datenmodell wird das Star-Schema aufgrund des ausgeprägten Know-how der Administratoren in vielen Anwendungen genutzt.

Typisch für die Art der Abfragen ist, dass normalerweise sehr viele Datensätze der Faktentabelle abgefragt werden, dass alle betroffenen Tabellen mit einem join verbunden werden und dass die Ergebnismenge im Vergleich zu den beteiligten Basisdatensätzen recht klein ist. Daher hat auch die Optimierung der Performance für die Generierung von Aggregaten im Star-Schema einen sehr hohen Stellenwert.<sup>127</sup>

Das Star-Schema ist insbesondere bei kleinen bis mittleren Datenmengen ein adäquater und leicht handhabbarer Modellierungsansatz. Doch trotz der Performance-Probleme bei sehr großen Modellen ist nach heutigem Stand der Technik dies die einzige gangbare Möglichkeit, Datenmengen in Terabyte und Petabyte-Größenordnungen in einem Data Warehouse zu speichern.<sup>128</sup>

---

<sup>124</sup> vgl. Totok (2000a), S. 176f.

<sup>125</sup> Moody, Kortink (2003), S. 11

<sup>126</sup> Holthuis (2001), S. 196

<sup>127</sup> Hahne (2002a), S. 101

<sup>128</sup> Hahne (2002a), S. 139

### 3.1.2 Multistar-Schema

In manchen Fällen ist eine Modellierung mit nur einer Faktentabelle nicht ausreichend. In komplexen Praxisfällen kommt es äußerst selten vor, dass alle Dimensionen in Abhängigkeit von nur einer Faktentabelle darstellbar sind. In diesem Fall ist ein Multistar-Schema verwendbar, welches in Abbildung 12 beispielhaft abgebildet ist.

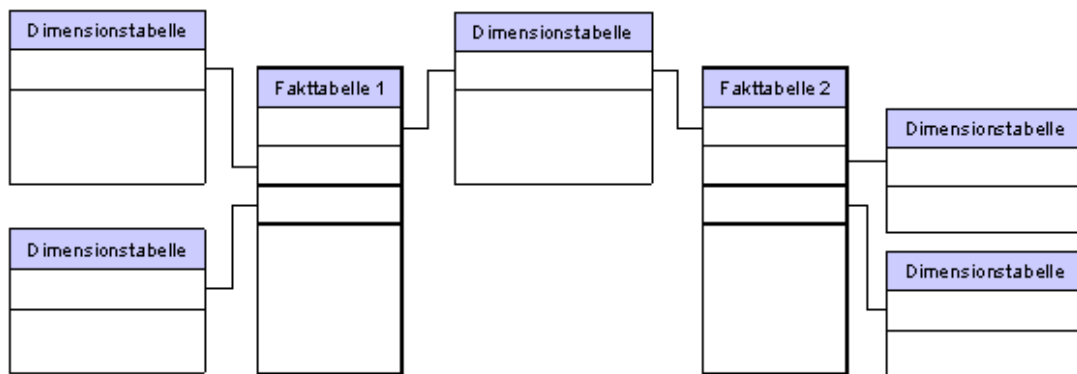


Abbildung 12: Multistar-Modell in einer Schemadarstellung

Wie in der Abbildung dargestellt ist, sind die Fakten teils durch die selben Dimensionstabellen beschrieben, teils werden unterschiedliche Dimensionstabellen zur Beschreibung verwendet.<sup>129</sup> Durch die erweiterte Anordnung aller Tabellen spricht man auch von einem Multi-Fakttabellenschema oder einer Galaxie.

In einem klassischen Star-Schema werden alle Fakten in nur einer einzigen Tabelle abgelegt. Gerade wenn viele Fakten sehr unterschiedlicher Dimensionierung in einem Data Warehouse zu speichern sind, bringt diese Speicherstruktur Nachteile mit sich. Da der Primärschlüssel in der Faktentabelle zusammengesetzt aus den Primärschlüsseln aller Dimensionstabellen ist, werden hier unnötig viele Null-Werte in den Dimensionstabellen benötigt, welche demzufolge bei Abfragen mit zu berücksichtigen sind. Eine naheliegende Verbesserung liegt in der Trennung in Faktentabellen, in welchen jeweils nur Fakten gleicher Dimensionierung gespeichert werden. Um die Semantik der einzelnen Fakten besser zu berücksichtigen, können darüber hinaus auch alle Fakten jeweils in einer eigenen Faktentabelle abgelegt werden. Dies erhöht aber im allgemeinen den Speicherbedarf.<sup>130</sup>

### 3.1.3 Snowflake-Schema

Wie bereits erläutert, ist die Besonderheiten beim Star- und Multistar-Schema, die denormalisierte Speicherstruktur. Das Snowflake-Schema bietet eine Möglichkeit, diese Option der Datenbanktechnologie bei Bedarf zu umgehen.

<sup>129</sup> vgl. Holthuis (2001), S. 197

<sup>130</sup> Hahne (2002a), S. 124

Um die großen Datenbestände in den Dimensionstabellen des Star-Schemas zu verringern, kann man durch Normalisierung einzelner, sehr großer Dimensionstabellen ein Star-Schema in ein sogenanntes Snowflake-Schema überführen.<sup>131</sup>

Die im Star-Schema auftretenden Redundanzen innerhalb der Dimensionstabellen sind bei manchen Modellierungsarbeiten nicht gewünscht oder nicht zweckmäßig. Aus diesen Erkenntnissen entstand das Snowflake-Modell. Dabei bleibt die Faktentabelle aus dem Star-Schema unverändert. An den Dimensionstabellen wird hingegen eine Normalisierung vorgenommen. Die Dimensionstabellen werden nach den Regeln der DBMS-Entwicklung normalisiert. Es entstehen dadurch mehrere Dimensionstabellen (siehe Abbildung 13). Durch die jeweilige Verknüpfung der Schlüssel der einzelnen Dimensionstabellen bleibt die (indirekte) Verbindung zur Faktentabelle erhalten.

Die Dimensionstabellen enthalten nicht mehr alle Dimensionselemente, sondern lediglich Daten über die Dimensionshierarchien, das heißt ein Tupel für jede Hierarchieebene. Die Dimensionstabellen sind über die Schlüsselattribute sowohl mit der zentralen Faktentabelle als auch mit den Attributtabellen, welche die deskriptiven Informationen über die Dimensionselemente enthalten, verknüpft.<sup>132</sup>

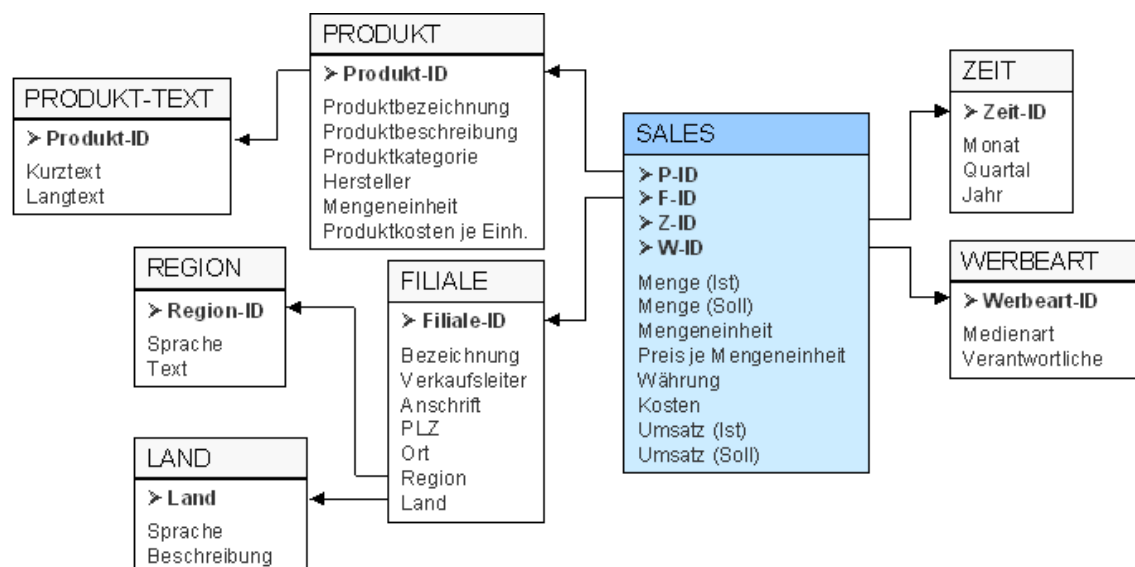


Abbildung 13: Snowflake-Schema aus der Fallstudie

Das Snowflake-Schema entspricht einem logischen, relationalen Modell, in dem eine Faktentabelle über Fremdschlüsselattribute mit Dimensionstabellen verbunden ist und die Dimensionen nicht vollständig denormalisiert sind.<sup>133</sup>

Im allgemeinen Ansatz des Snowflake-Schemas werden alle Kombinationen von Aggregationen jeweils in eigenen Faktentabellen gespeichert. Mit geeigneten Tools kann

<sup>131</sup> Holthuis (2001), S. 202

<sup>132</sup> Holthuis (2001), S. 203

dies vermieden werden, so dass hinsichtlich Performance und Speicher-Nutzung ein guter Kompromiss erreicht werden kann, da explizit angegeben werden kann, welche Aggregationen in eigenen Faktentabellen hinterlegt und welche „on-the-fly“ berechnet werden sollen.<sup>134</sup>

Das Snowflake-Schema stellt eine Erweiterung des Star-Schemas dar, bei dem die Dimensionen stärker oder ganz normalisiert werden. Die Anwendung des Snowflake-Schemas ist dann geboten, wenn die Dimensionsebenen sehr ungleichmäßig besetzt sind.<sup>135</sup> Da die grafische Darstellung entfernt an eine Schneeflocke erinnert, nennt man dieses Entwurfsmuster Snowflake-Schema.<sup>136</sup>

Das Snowflake-Schema ist bezüglich der funktionalen Abhängigkeiten, die durch die Klassifikationsbeziehungen definiert werden (z.B. Produktgruppe → Produktfamilie), normalisiert. Dadurch ergeben sich die üblichen Vorteile einer normalisierten Struktur, vor allem die Vermeidung von Änderungsanomalien.<sup>137</sup>

Durch die Vermeidung der Redundanzen wird weniger Speicherbedarf benötigt. Dies ist ein Vorteil des Snowflake-Schemas. Allerdings entsteht bei diesem Modell durch die neuen Schlüsselmerkmale, man spricht auch von einer erhöhten Verknüpfungstiefe, der Dimensionstabellen ein erhöhter Verwaltungsaufwand. Innerhalb dieser Struktur ergeben sich teilweise kürzere Zugriffszeiten auf die Daten. In der Praxis werden häufig nur große Dimensionstabellen normalisiert. Dies ist als *Partial Snowflake-Schemata* bekannt.<sup>138</sup>

Gegenüber dem Star-Schema liegt der Nachteil des Snowflake-Schemas in der höheren Komplexität. Dies erschwert es, durch die Snowflake-Struktur zu navigieren. Dieser Nachteil tritt insbesondere dann auf, wenn traditionelle Abfragetools, eingesetzt werden.

Leistungsverbesserungen und Speicherplatzeinsparungen durch ein Snowflake-Design wiegen den Nachteil der höheren Komplexität nur dann auf, wenn die Dimensionstabellen sehr groß sind und auch sehr viele Attribute auf niedrigen Ebenen der Dimensionshierarchie existieren.<sup>139</sup>

Der wesentliche Nachteil von Snowflake-Schemas, so wird in der Literatur immer wieder angeführt, ist die im Vergleich zum klassischen Star-Schema hohe Komplexität, welche Abfragen an das Data Warehouse schwer formulierbar und fehleranfällig werden lassen. Diese Problematik kann mit Tools, welche auf semantischer Ebene die

---

<sup>133</sup> Lehmann (2001), S. 76

<sup>134</sup> Hahne (2002a), S. 150

<sup>135</sup> Totok (2000a), S. 177

<sup>136</sup> Bauer, Günzel (2001), S. 200

<sup>137</sup> Bauer, Günzel (2001), S. 200f.

<sup>138</sup> vgl. Schinzer, Bange, Mertens (1999), S. 50f.

<sup>139</sup> Behme, Holthuis, Mucksch (2000), S. 229

Verwaltung des Data Warehouse ermöglichen und gleichzeitig diese semantische Schicht dem Endanwender-Tool transparent darstellen, zum Teil umgangen werden.<sup>140</sup>

### 3.1.4 Eine Gegenüberstellung von Star-Schema und Snowflake-Schema

Wie man erkennen kann, ist die Faktentabelle im Star-Schema weiterhin normalisiert, während die Dimensionstabellen bewusst gegen die Normalisierung verstoßen. Dies führt zu gewissen Redundanzen innerhalb der Dimensionstabellen.<sup>141</sup>

Tabelle 4: Star-Schema und Snowflake-Schema im Vergleich

Star-Schema	Snowflake-Schema
Fakttabelle ist normalisiert	Tabellen sind (vollkommen) normalisiert
Dimensionstabellen sind denormalisiert abgespeichert	Jede Verdichtungsebene einer Dimension wird in einer relationalen Datenbank-Tabelle gespeichert
Geringe Anzahl Tabellen	Höhere Anzahl Tabellen
Keine aufwändige join-Verbindungen bei Abfragen	Durch Normalisierung entstehen relativ kleine Dimensionstabellen (Lookup-Tabellen)
Einfache Struktur durch Denormalisierung	Komplexe Datenstrukturen
Kleinerer Wartungsaufwand	Hoher Verwaltungsaufwand
Aggregationsbildung ist schwierig	Aggregationsbildung ist optimal
Grad der Normalisierung	Browsing-Funktionalitäten
Schnellerer Datenzugriff	Geringerer Speicherplatzbedarf
Klassifikation als eine Tabelle abgebildet	Tabellenname entspricht dem Namen der Klassifikationsstufe
Oberbegriff als Namenskonvention der Fakttabelle	Fremdschlüssel der Fakttabelle sind in der niedrigsten Granularität

Eine allgemeine Aussage, ob Star- oder Snowflake-Schemata besser geeignet sind, ist nicht sinnvoll, da die Vorteile des Star-Schemas, wie schnellerer Datenzugriff, und die Vorteile des Snowflake-Schemas, wie geringerer Speicherplatzbedarf und bessere Änderungsfreundlichkeit, stark von den konkreten Daten- und Anfragecharakteristiken abhängig sind. Daher werden in der Praxis oft Mischformen verwendet.<sup>142</sup>

<sup>140</sup> Hahne (2002a), S. 150

<sup>141</sup> Bauer, Günzel (2001), S. 202

<sup>142</sup> Bauer, Günzel (2001), S. 204

### 3.1.5 Zusammenfassung und Fazit

Das Star- und Snowflake-Schema bieten je nach Lösungsanforderung verschiedene Vor- oder Nachteile. In der Tabelle 4 sind diese Merkmale der Modelle gegenüber gestellt. Da keines der beiden Schemen entscheidende Vorteile aufweist, die Strukturen aber sehr ähnlich sind, liegt es nahe die beiden Modelle in der Praxis zu kombinieren.

Je nach Anforderung der Anwendung werden hierzu einzelne Dimensionen normalisiert und andere Dimensionen, wie im Star-Schema, in denormalisierter Form modelliert. Welche Dimensionen denormalisiert gespeichert werden sollen, ist jeweils nach der entsprechenden Anforderung bei der Modellierung zu entscheiden. Als Entscheidungsgrundlage für die Speicherform sind in die Überlegungen die *Frequenz der Änderung*, die *Anzahl der Dimensionselemente*, die *Anzahl der Klassifikationsstufen innerhalb einer Dimension* und die *Materialisierung von Aggregaten für Dimensionsstufen* einzubeziehen.<sup>143</sup>

Um mit multidimensionalen Datenstrukturen messbare Leistungsvorteile erzielen zu können, macht es Sinn, die Daten multidimensional in einer multidimensionalen Matrix zu speichern. Multidimensionale Datenbanken basieren auf der Annahme, dass alle Kombinationen an Dimensionen vorkommen, und erstellen „im voraus“ basierend auf dem Kreuzprodukt aller Wertebereiche der Dimensionsobjekte entsprechende Matrizen. Die Adresse jeder Zelle berechnet sich dann durch eine simple Kette von Additionen und Multiplikationen unter Verwendung der dimensionalen Werte und der Größe jedes einzelnen Dimensionselementes.<sup>144</sup>

## 3.2 Multidimensionale Datenbanken

Eine der Philosophien des multidimensionalen Modells ist es allerdings, auf die universelle Anwendbarkeit zu verzichten und dadurch innerhalb der multidimensionalen Analyse numerischer Daten einen höheren Automatisierungsgrad zu erreichen. Zu diesem Zweck integriert das multidimensionale Datenmodell Konzepte wie Klassifikationshierarchien, die eine spezielle Semantik tragen, aber in dem ausgewählten Anwendungsgebiet eine wichtige Rolle spielen.

Eine konzeptuelle Modellierungsmethode, die zur Abbildung multidimensionaler Datenmodelle geeignet sein soll, muss also diese speziellen semantischen Konstrukte des multidimensionalen Datenmodells (bzw. Paradigmas) berücksichtigen. [...] Ein wichtiges Merkmal des multidimensionalen Datenmodells ist die Unterscheidung von Klassifikationsstufen, beschreibenden Attributen und Kenngrößen. Diese Unterscheidung ist in einem E/R-Modell nicht direkt ersichtlich.<sup>145</sup>

---

<sup>143</sup> vgl. Bauer, Günzel (2001), S. 204

<sup>144</sup> Lehmann (2001), S. 41

<sup>145</sup> Bauer, Günzel (2001), S. 157f.



Innerhalb der physisch multidimensionalen Datenbanksysteme gibt es verschiedene Speicher-Ansätze der kommerziellen Anbieter. Zum einen wird ein sog. Hypercube (Hyper-Würfel) Speichermodus verfolgt. Dabei werden alle Daten in nur einer multidimensionalen Matrix, dem Hypercube, gespeichert. Bei dünn besiedelten Matrizen entsteht hier eine Verschlechterung der Performance sowie der Replikationsdauer, da Speicherplatz ungenutzt bleibt. Der zweite Ansatz verfolgt die Datenspeicherung in mehreren Matrizen. Diese meist mehrdimensionalen Cubes werden untereinander in Beziehung gesetzt. Bei diesem Multicube-Ansatz ist die Verknüpfung der Matrizen ein Nachteil bei der Abfragegeschwindigkeit.<sup>146</sup>

Multidimensionale Datenbanken speichern Daten physikalisch multidimensional in Array- (Zell-) Strukturen, die direkt adressiert werden können. Durch diese Architektur eignen sich multidimensionale Datenbankmanagementsysteme (MDBMS) besonders gut zur Datenanalyse, da die besten Ergebnisse bezüglich der Geschwindigkeit beim Datenzugriff und mehrdimensionaler Kalkulationen erzielt werden.<sup>147</sup> Der Hauptvorteil liegt in der Performance bei Datenmanipulationen. Allerdings ist die Datenspeicherung derzeit auf ungefähr 20 Gigabyte begrenzt. Bei umfangreichen und verschiedenen Dimensionen ist diese Größe durchaus zu erreichen.<sup>148</sup>

### 3.2.1 On-Line Analytical Processing (OLAP)

Unter OLAP wird per Definition ein Konzept zur analytischen multidimensionalen Datenauswertung für die Unternehmensführung und der unterstützenden Organe verstanden. Die Bezeichnung des „On-Line“ steht dabei für eine schnelle Abfragegeschwindigkeit. Diese soll an die Systemabfragen operativer Datenbanken heran kommen.

E. F. Codd et al. begann 1993 eine Diskussion über die Schwächen relationaler Datenbanksysteme multidimensionaler Analysen. Aus dieser Erkenntnis entwarf er einen Katalog mit 12 Grundregeln für Analysewerkzeuge.<sup>149</sup> Diese sind bis heute, unter dem OLAP-Schlagwort, in nahezu jedem analytischen Informationssystem enthalten.

Hinter dem Begriff OLAP verbirgt sich die Idee, die in der Datenbank gespeicherten Daten analysieren zu können. Dabei wird eine Unterstützung bei Anfragen für Analyse-zwecke und die Aufbereitung von Geschäftsdaten für Entscheider in einem Unternehmen über OLAP-Tools realisiert.

An die OLAP-Tools werden die Anforderungen folgender Funktionalitäten gestellt:<sup>150</sup>

1. Datenrepräsentation in aggregierter oder summierter Form
2. Grad der Aggregation ist variabel

---

<sup>146</sup> vgl. Behme, Holthuis, Mucksch (2000), S. 216ff.

<sup>147</sup> Schinzer, Bange (1999), S. 54

<sup>148</sup> vgl. Wieken (1999), S. 89

<sup>149</sup> vgl. Totok (2000a), S. 55

<sup>150</sup> vgl. Vossen (1999), S. 679

3. mehrdimensionale Sicht auf die Daten
4. interaktive Analyseformen
5. Datenbestand kann sehr großes Volumen besitzen (im GB- bis TB-Bereich).

### 3.2.2 Relationales OLAP (ROLAP)

Die bewährte Technologie der relationalen Datenbanken kann durch Erweiterungen der Softwareanwendungen, Server- und Endbenutzerseitig, für multidimensionale Sichtweisen ergänzt werden. In diesem Zusammenhang wird dann von der ROLAP-Technologie oder der virtuellen Multidimensionalität gesprochen.<sup>151</sup>

Wenn das speichernde System die Datenhaltung der zu analysierenden Daten in Relationen vornimmt, spricht man von relationalem OLAP (ROLAP). Ein ROLAP-Server übernimmt die Datenbevorratung, kann dabei zur Client-Entlastung aber auch Berechnungs- und Transformationsfunktionen übernehmen.

Der Einsatz relationaler Datenbanksysteme in Verbindung mit multidimensionalen Auswertungswerkzeugen wird auch als Relationales OLAP (=ROLAP) bezeichnet.<sup>152</sup>

Relationale Datenbanken haben sich vor allem als Datenspeicher sehr großer Datenvolumen in der Praxis etabliert. Mehrere Terabyte sind hier keine Seltenheit. Um eine relationale Datenbank um die ROLAP-Funktionen zu erweitern, wird eine dreistufige Rechnerarchitektur benötigt. Zwischen die Datenbank und die OLAP-Funktionen (Business Intelligence Tools) wird eine OLAP-Engine als Schnittstelle implementiert. Die ROLAP-Engine bereitet die vorliegenden relationalen Daten in multidimensionale Strukturen auf, wodurch ein effizienter Zugriff der Anfragen realisiert wird.<sup>153</sup>

### 3.2.3 Multidimensionales OLAP (MOLAP)

Es werden aber auch multidimensionale Datenbanken eingesetzt, welche „hinsichtlich der physikalischen Speichertechnik auf die OLAP-Denkweise ausgerichtet sind, um zusätzliche Geschwindigkeitsvorteile zu aktivieren.“<sup>154</sup>

Sofern die Datenhaltung der zu analysierenden Daten in multidimensionaler Form vornimmt, spricht man hier von MOLAP.

Beim multidimensionalen OLAP (MOLAP) werden die Daten bereits physikalisch in multidimensionalen Strukturen gespeichert, weshalb eine weitere Aufbereitung vor dem Zugriff der Analysewerkzeuge nicht mehr notwendig ist. Die Möglichkeit des direkten Zugriffs auf Dimensionselemente und voraggregierte Dimensionshierarchien in multidimensionalen Datenbanken sorgen für sehr schnelle Antwortzeiten bei Abfrage und Kalkulation.<sup>155</sup> Die Daten werden also bereits vor der Nutzung aus der relationalen Da-

---

<sup>151</sup> vgl. Chamoni, Gluchowski (2000), S. 344

<sup>152</sup> Mucksch, Behme (2000), S. 78

<sup>153</sup> vgl. Schinzer, Bange (1999), S. 54

<sup>154</sup> Chamoni, Gluchowski (2000), S. 344

<sup>155</sup> Schinzer, Bange (1999), S. 57

tenbank ausgelesen und in der mehrdimensionalen Datenbank in die Zellen eingelesen.<sup>156</sup>

### 3.2.4 Hybrides OLAP (HOLAP)

Die als hybrides OLAP (HOLAP) bezeichnete Kombination von multidimensionaler und relationaler Speicherung verbindet die Vorteile von MOLAP und ROLAP.<sup>157</sup>

Abhängig von den Erfordernissen der Anwendung bezüglich Abfragehäufigkeit und Aggregationsstufen der Daten wird ein Teil der Daten physikalisch multidimensional gespeichert, während ein anderer (meist größerer) Teil der Daten in einer relationalen Datenbank vorgehalten und nur zur Laufzeit multidimensional aufbereitet wird.<sup>158</sup>

### 3.2.5 Ein Vergleich der OLAP-Architekturen

Der Microsoft SQL Server gehört zu den physikalischen Data Warehouses. Dieser Server kennt die ROLAP, MOLAP und HOLAP Anwendungen. Diese verschiedenen Speichervarianten werden in Kapitel 4.2 in Bezug auf den Microsoft SQL Server nochmals aufgegriffen. In der Tabelle werden die Vor- und Nachteile der Varianten gegenübergestellt.

Tabelle 5: OLAP-Architekturen im Vergleich<sup>159</sup>

Architektur	Vorteile	Nachteile
<b>ROLAP</b>	<ul style="list-style-type: none"> <li>▪ Robuste Datenbanktechnologie</li> <li>▪ Standard-Abfragesprache (SQL)</li> <li>▪ Einfacher Datenimport via SQL</li> <li>▪ Sicherheitsmechanismen auf relationaler Ebene</li> <li>▪ Große Datenmengen möglich</li> <li>▪ Am häufigsten für DWH verwendet</li> </ul>	<ul style="list-style-type: none"> <li>▪ Standard-SQL für OLAP nicht ausreichend</li> </ul>
<b>MOLAP</b>	<ul style="list-style-type: none"> <li>▪ Sehr gute Antwortzeiten bei kleinen Datenmengen (&lt; 1 Gbyte)</li> <li>▪ Intuitive Abfragesprache</li> <li>▪ Einfaches Hinzufügen von Dimensionen und Aggregationsstufen</li> </ul>	<ul style="list-style-type: none"> <li>▪ Problematik dünnbesetzter Würfel</li> <li>▪ Proprietäre DBMS und bisher keine Standard-Abfragesprache</li> <li>▪ Schlechte Antwortzeiten bei größeren Datenvolumina</li> <li>▪ Zur Integration externer Quellen Transformation zu SQL nötig</li> <li>▪ Sinkende Effizienz bei Zugriff auf persistentes Speichermedium</li> </ul>

<sup>156</sup> vgl. Wieken (1999), S. 83

<sup>157</sup> Schinzer, Bange (1999), S. 56

<sup>158</sup> Schinzer, Bange (1999), S. 57

<sup>159</sup> Schmidt-Thieme (2002)

<b>HOLAP</b>	<ul style="list-style-type: none"> <li>▪ Vereinte Vorteile von ROLAP und MOLAP</li> <li>▪ Multidimensionales DB-System greift nicht mehr direkt auf operative Systeme zu, sondern auf die Detaildaten im relationalen Teil des DWH</li> <li>▪ Nur geringe Anzahl persistenter Aggregationen nötig</li> </ul>	<ul style="list-style-type: none"> <li>▪ Höherer Implementations- und Wartungsaufwand</li> <li>▪ Keine einheitliche Abfragesprache</li> </ul>
--------------	--	---

### 3.2.6 Zusammenfassung und Fazit

Der Inhalt eines Data Warehouse wird maßgeblich durch das semantische, multidimensionale Datenmodell bestimmt.<sup>160</sup>

Für die Datenhaltung in Data Warehouse-Systemen gibt es nach heutigem Stand der Technik zwei Speichermöglichkeiten. Dies ist zum einen die relationale und zum anderen die multidimensionale Datenhaltung. Bei der multidimensionalen Speicherung sind die Zellen der Datenwürfel direkt vom System adressierbar. Entscheidungsrelevante Daten können in diesem System gut berechnet und analysiert werden. Aus diesem Grund spielt die multidimensionale Datenhaltung im Entwicklungsbereich der innovativen Konzepte für Data Warehouses eine entscheidende Rolle.

## 3.3 Modellierung von Zeit

Die „Zeit“ (engl. *time*) ist für beinahe jede Fragestellung im Management eine relevante Bezugsgröße. Dadurch kommt ihr auch im Data Warehousing eine sehr wichtige Rolle zu. Dies ist nicht zuletzt durch die besondere Stellung der Zeitdimension in multidimensionalen Modellen ersichtlich.

Die Zeitdimension hat in fast allen Data Warehouse-Datenmodellen eine herausragende Bedeutung, da die zeitliche Qualifizierung des betrachteten Zahlenmaterials essenziell ist.<sup>161</sup>

Es gibt unterschiedliche Möglichkeiten eine Einteilung der Datenbanksysteme über die Zeit vorzunehmen. Hierzu müssen zu Beginn zwei Begriffe erklärt werden. Als *Instant* (*Instanz*) wird ein fixer Zeitpunkt der Zeitachse bezeichnet und die Zeit zwischen zwei „instants“ (Fixzeitpunkten) nennt man *time interval* (*Zeitintervall*). Datenbanksysteme, welche mindestens *transaction time* oder *valid time* berücksichtigen, werden als „Temporale Datenbanken“ bezeichnet. Unter einer *transaction time* versteht man in diesem Zusammenhang eine *Zeitraumbezogene Betrachtung* der Datenbank. Unter *valid time*

<sup>160</sup> Lehmann (2001), S. 158

<sup>161</sup> Hahne (2002a), S. 86

werden stichhaltige Zeitpunkte oder -verläufe, Betrachtungen mit einer bestimmten Zuordnung, zusammengefasst.<sup>162</sup>

Nach diesen Erklärungen kann die folgende Einteilung von Datenbanksystemen vorgenommen werden:

Tabelle 6: Datenbankklassifikation<sup>163</sup>

	no valid time	valid time
no transaction time	snapshot database	historical database
transaction time	rollback database	bitemporal database

### Valid time

Verschiedene Dimensionen unterliegen einem unterschiedlich schnellen Wandel. Bei vielen Kennzahlen der Dimensionen (z.B. Umsatz, etc.) entstehen täglich neue Werte. Andere wiederum sind nur einem seltenen Wandel ausgesetzt, es gibt jedoch nur sehr wenige Dimensionen welche als statisch betrachtet werden können. Für die eher selten wechselnden Dimensionen (z.B. Produkt) bildete Kimball den Begriff der „*slowly changing dimensions*“.

Dazu schreibt Kimball:

*“Slowly changing dimension technique allows us to see new and historical fact data by either the new or prior attribute values.”*

Zeitpunktbezug und Zeitraumbezug sind unterscheidbar: Nach diesem Muster werden also einerseits stichtagsbezogene Größen abgebildet (z.B. Lagerbestand am Inventurtag), andererseits werden die Geschäftsentwicklungen in einem Zeitraum (z.B. Umsatzentwicklung im ersten Quartal) analysierbar.<sup>164</sup>

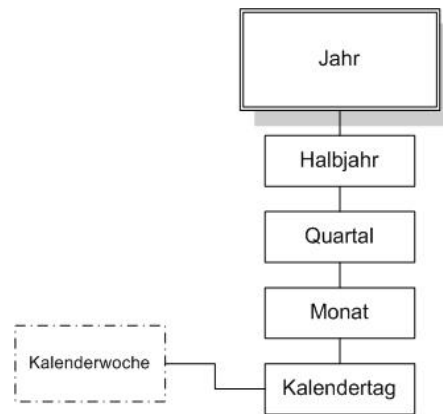
Die Hierarchie einer Zeitdimension lässt sich, ausgehend vom Kalenderjahr als höchster Verdichtungsstufe, über Halbjahre, Quartal, Monate hin zu den einzelnen Kalendertagen als niedrigster Verdichtungsebene aufspannen. [...] Kommt jedoch beispielsweise die für Handelsunternehmen typische Berücksichtigung einzelner Kalenderwochen hinzu, ist eine eindeutige hierarchische Abbildung nicht mehr möglich.<sup>165</sup>

<sup>162</sup> vgl. Hahne (2003), S. 1f.

<sup>163</sup> Hahne (2003), S. 2

<sup>164</sup> vgl. Kimball (1996), S. 95ff.

<sup>165</sup> Holthuis (2000), S. 179

Abbildung 14: Zeithierarchie<sup>166</sup>

In diesem Beispiel der Dimension „Zeit“ als Instanz wird ersichtlich, dass die Quartale eines Jahres der selben Dimension wie die Monate zugeordnet werden. Die Quartale sind allerdings in einer anderen Aggregation gespeichert. Dies kann auf alle Dimensionen mit unterschiedlichen Aggregationsstufen übertragen werden, es handelt sich bei diesem Phänomen um keine Zeitspezifische Besonderheit

Eine Schwierigkeit der Dimension „Zeit“ wird in der exemplarischen Dimensionstabelle in Tabelle 7 aufgeschlüsselt. Darin wird die Dimension Zeit für vier Tage dargestellt. Bereits dieser Tabellenausschnitt nimmt eine komplexe Speicherstruktur ein. Stellt man sich diese Tabelle für mehrere Jahre im Verlauf vor, so ist ein großer Datenumfang leicht nachvollziehbar. Aufgrund der Wichtigkeit dieser Dimension muss der Aufwand jedoch entsprechend betrieben werden.

*Data warehouses always need an explicit date dimension table. There are many date attributes not supported by the SQL date function, including fiscal periods, seasons, holidays, and weekends. Rather than attempting to determine these non-standard calendar calculations in a query, we should look them up in a date dimension table.*<sup>167</sup>

Tabelle 7: Beispiel einer Dimensionstabelle der Dimension „Zeit“<sup>168</sup>

Datum ID	Datum	Volldarstellung Datum	Wochentag	Kalendermonat	Kalenderjahr	Fiskal-darstellung	Feiertag-indikator	Wochentag-indikator
1	01.01.04	01. Januar 2004	Donnerstag	Januar	2004	F2004-01	Feiertag	Wochentag
2	02.01.04	02. Januar 2004	Freitag	Januar	2004	F2004-01	Kein-Feiertag	Wochentag
3	03.01.04	03. Januar 2004	Samstag	Januar	2004	F2004-01	Kein-Feiertag	Wochenende
4	04.01.04	04. Januar 2004	Sonntag	Januar	2004	F2004-01	Kein-Feiertag	Wochenende

<sup>166</sup> Holthuis (2000), S. 179<sup>167</sup> Kimball (1996), S. 41<sup>168</sup> vgl. Kimball (1996), S. 41

Oftmals wird im Entwurf mehrdimensionaler Modelle die Zeitdimension auch in zwei getrennten Dimensionen modelliert. In der einen Dimension werden nur die Jahre angesprochen und in der anderen alle anderen Elemente bis zur Monats- und Quartals-ebene und die Verdichtung über alle Monate bzw. Quartale. Der Vorteil dieser getrennten Modellierung liegt in der Administration der Zeitdimension. Die Erweiterung der Dimension um ein Jahr bedeutet im Fall nur einer Dimension, dass für das hinzukommende Jahr Dimensionselemente für die Monate mit den Attributen für das Quartal und das Jahr angelegt werden müssen. Im zweiten Fall braucht lediglich die Dimension Jahr um ein weiteres Element erweitert werden.<sup>169</sup>

Die vorhandene Dynamik, durch sich ändernde Merkmale aller Dimensionen erfordern bei der Dimension „Zeit“ besondere Datenmodellierungsmethoden. Da im Data Warehouse alle Dimensionen in einer Affinität zu der „Zeit“-Dimension stehen, besitzen eben alle Änderungen auch eine mehr oder weniger tiefe Auswirkung auf die „Zeit“. Bei Saisonspezifischen Dimensionen (z.B. saisonale Entwicklung, Lebenszyklusmodelle) muss bei der Modellierung auf eine unterschiedliche Dynamik der Datenwerte geachtet werden. In bestimmten Zeiten sind ständig aktuelle Werte erforderlich, hingegen außerhalb der Saison kann eine Kennzahl auch über Monate den selben Wert besitzen. In diesem Fall braucht das System nicht unnötig belastet zu werden.

Bei der Modellierung eines multidimensionalen Datenwürfels muss unter anderem auch auf die inhärente Zeitdimension die Granularität festgelegt werden, in der die Daten vorgehalten werden sollen. Die Festlegung der Zeitgranularität erfolgt jedoch abhängig von der Zeitgranularität der operativen Vorsysteme. Die Betrachtung diskreter Zeiteinheiten und die Definition der Zeitgranularität haben sowohl die multidimensionale Modellierung als auch die temporale Datenhaltung gemeinsam.<sup>170</sup>

### Fazit

Der Aspekt der hierarchischen Dimensionsmodelle ist ein wichtiges Thema bei der Entwicklung analytischer Informationssysteme und dies insbesondere bei Data Warehouses, wo verschiedene Typen hierarchischer Strukturen gebildet werden.<sup>171</sup>

Abschließend ist festzuhalten, dass die universelle Relevanz von Zeitaspekten und deren Vielfältigkeit eine nicht triviale Problematik bei der Datenmodellierung darstellen und somit stets der besonderen Beachtung bedürfen.<sup>172</sup>

---

<sup>169</sup> Hahne (2002a), S. 87

<sup>170</sup> Bauer, Günzel (2001), S. 184

<sup>171</sup> Hahne (2003), S. 6

<sup>172</sup> Holthuis (2000), S. 180

### 3.4 Vom mehrdimensionalen Entity-Relationship-Modell zur Methode „Hahne“

Wie bereits erwähnt stehen auf der semantischen Modellierungsebene verschiedene graphische Modellierungsmöglichkeiten zur Verfügung. Für die logische Modellebene stehen jedoch bis heute keine einheitlich standardisierte Modellierungsmöglichkeit der grafischen Darstellung zur Verfügung. Michael Hahne widmet sich seit geraumer Zeit in seinen Studien diesem Thema unter den Modellierungsaspekten für das SAP Business Information Warehouse. Dabei entwickelte er eine Schablonen-Datei auf Basis Microsoft Visio zur grafischen Darstellungsmöglichkeit des erweiterten Star-Schemas der SAP AG<sup>173</sup>.

Eine Werkzeuggestützte Modellierung basiert auf der Möglichkeit, ausgehend von der semantischen Modellebene losgelöst von möglichen Zielplattformen auf Ebene des Fachkonzepts abzubilden sowie dieses Modell dann auf die weiteren Ebenen herunter zu brechen und in festgelegten Datenbanksystem-Strukturen zu generieren.<sup>174</sup>

Der Datenhaltung im SAP Business Information Warehouse liegt eine relationale Datenbankarchitektur zugrunde. Die Modellierung der mehrdimensionalen Datenhaltung für das SAP-System wird für gewöhnlich in dem bekannten Star-Schema vorgenommen. Allerdings gibt es hierzu bestimmte Ergänzungen, wodurch ein „erweitertes Star-Schema“ für das SAP Business Information Warehouse entwickelt wurde. Die Dimensionstabellen bilden dabei die Verbindungsknoten, in Form der Tabelle, zwischen der Faktentabelle und den für das Schema eingeführten Stammdatentabellen. In diesem Schema wird das Modell deutlich komplexer als das klassische Star-Schema. Im Business Warehouse der SAP AG wird eine strikte Unterteilung in Stammdaten und Bewegungsdaten vorgenommen. Dies ist darin begründet, dass die Stammdaten Info-Cube übergreifend verwendet werden können, hingegen die Bewegungsdaten lediglich einem Info-Cube zur Verfügung stehen.<sup>175</sup>

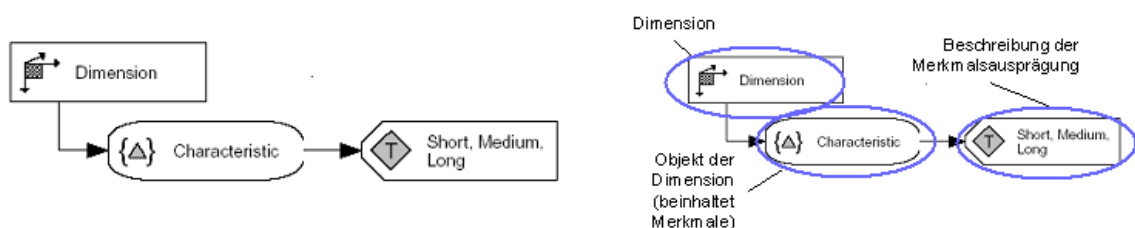


Abbildung 15: Dimension mit einem Merkmal und deren Beschreibungen<sup>176</sup>

<sup>173</sup> SAP AG ist die Abkürzung der Aktiengesellschaft Systeme, Anwendungen, Produkte in der Datenverarbeitung

<sup>174</sup> Hahne (2004a), S. 2

<sup>175</sup> vgl. Hahne (2004a), S. 1ff.

<sup>176</sup> vgl. Hahne (2004a), S. 8



Wichtigster Typ eines Info-Cubes ist der Basis-Cube, dessen Darstellung auf dem Objekt *Basis-Cube* basiert. Um den Cube herum werden die einzelnen Dimensionen angeordnet (siehe Abbildung 16).<sup>177</sup>

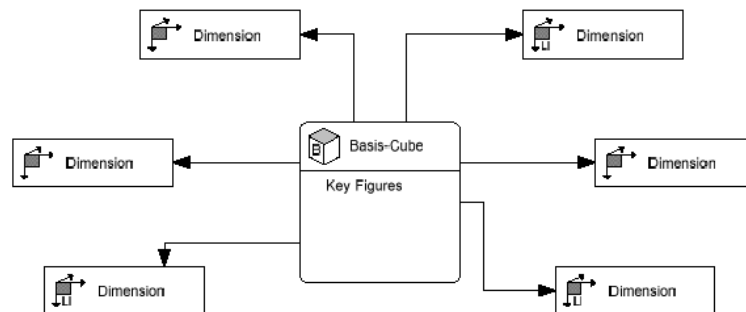


Abbildung 16: Modellierung von Basis-Cubes<sup>178</sup>

<sup>177</sup> Hahne (2004a), S. 10

<sup>178</sup> Hahne (2004a), S. 10

## 4 Architektur und Arbeitsweise des Microsoft SQL Server 2005

Die SQL Server-Gruppe von Microsoft gehört durch die erweiterten Anwendungswerkzeuge nicht nur zu den Daten-Speichersystemen, sondern wird aufgrund der Erweiterungen zu den Business Intelligence Werkzeugen gezählt werden. Nach den allgemein akzeptierten Definitionen besitzt der Microsoft SQL Server 2005 (Codename „Yukon“) alle Funktionen eines Data Warehouses.

Als deskriptive Anfragesprache für das Datenmodell steht SQL zur Verfügung, welches auf der relationalen Algebra basiert.<sup>179</sup>

Mit dem Begriff **Data Warehouse** i.e.S. wird generell eine von den operationalen DV-Systemen isolierte Datenbank umschrieben, die als **unternehmensweite Datenbasis** für alle Ausprägungen managementunterstützender Systeme dient und durch eine strikte Trennung von operationalen und entscheidungsunterstützenden Daten und Systemen gekennzeichnet ist.<sup>180</sup>

Das Data Warehouse kann grundsätzlich irgendein Datenbanksystem sein, das für die Verarbeitung sehr großer Datenmengen geeignet ist. In der Praxis sind dies meistens *relationale* oder *multidimensionale* Datenbanken, die speziell für die Verarbeitung von Warehouse-Daten erweitert sind.<sup>181</sup>

Über die beschriebenen multidimensionalen Datenstrukturen sind flexible Ad-hoc-Abfragen und Berichte relativ leicht zu realisieren. Die intuitive Gestaltung der Abfragen und der somit einfachen Überprüfung der aufgestellten Thesen in verschiedenen Unternehmensbereichen, ist eine der Stärken der multidimensionalen Datenstrukturen im Vergleich zu Informationssystemen der Historie.<sup>182</sup>

### 4.1 Die Data Warehouse Referenzarchitektur und die Microsoft SQL Server 2005 Architektur

In diesem Kapitel wird die Architektur des SQL Servers vorgestellt und mit der Referenzarchitektur in Verbindung gebracht.

Der Anwender soll durch ein Data Warehouse in die Lage versetzt werden, in Form von Daten repräsentierte Informationen in Wissen umzusetzen, um daraus langfristige Strategien, aber auch konkrete Aktionen abzuleiten. Das Data-Warehouse-Konzept

---

<sup>179</sup> Bauer, Günzel (2001), S. 198

<sup>180</sup> Mucksch, Behme (2000), S. 6

<sup>181</sup> Lehmann (2001), S. 41

<sup>182</sup> vgl. Holthuis (2000), S. 152

wird daher auch als Instrument zur strategischen Unternehmenssteuerung propagiert.<sup>183</sup>

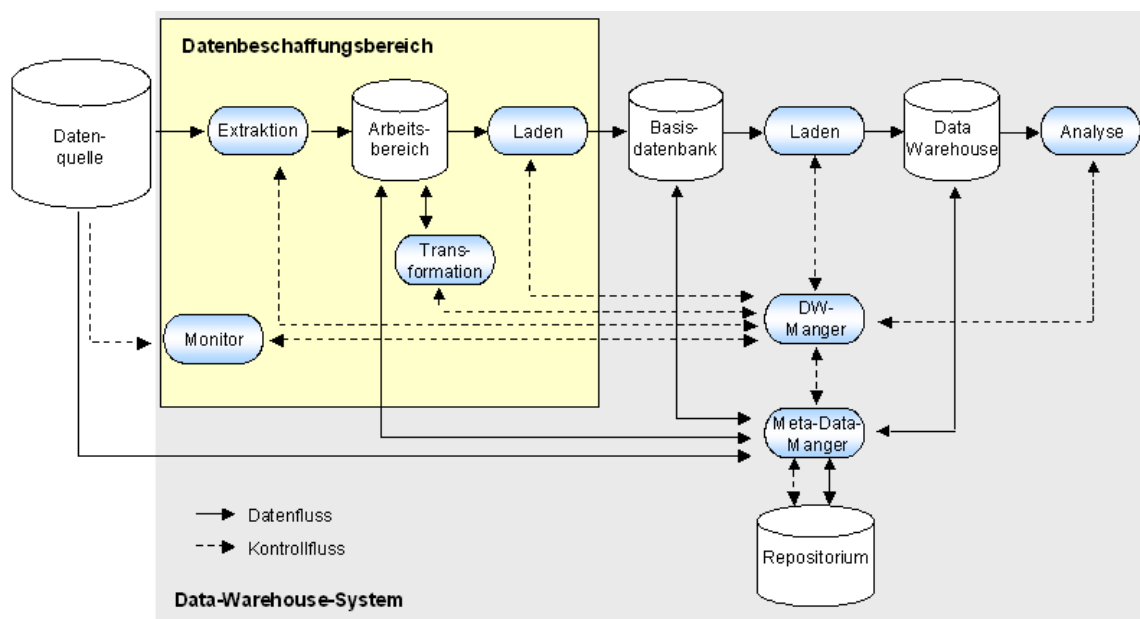


Abbildung 17: Referenzarchitektur für ein Data Warehouse-System<sup>184</sup>

Ziel des Architekturmodells ist es, eine vollständige, konsistente und jederzeit aktuelle Dokumentation über die Objekte des Warehouse zu erhalten.<sup>185</sup>

Die Speicherung von multidimensionalen Strukturen in einem relationalen Datenmodell ist in manchen Fällen nicht ganz unproblematisch.

Ziel der relationalen Abbildung ist die multidimensionale Speicherung so abzubilden, dass:<sup>186</sup>

- möglichst wenig anwendungsbezogene Semantik, die im multidimensionalen Modell enthalten ist, bei der Abbildung verloren geht (z.B. Klassifikationshierarchien),
- die Übersetzung von multidimensionalen Anfragen effizient geschehen kann,
- die Abarbeitung der übersetzten Anfragen durch das relationale Datenbanksystem auf möglichst effiziente Weise erfolgt und
- die Wartung (z.B. das Laden der neuen Daten) der entstandenen Tabellen einfach und schnell erledigt werden kann.

<sup>183</sup> Totok (2000a), S. 3

<sup>184</sup> Bauer, Günzel (2001), S. 36

<sup>185</sup> Lehmann (2001), S. 154

<sup>186</sup> Bauer, Günzel (2001), S. 198

### Data Warehouse – Anwendung und Wartung

Die Struktur eines Data Warehouses kann allgemein wie folgt beschrieben werden:<sup>187</sup>

1. Die Datenquellen sind relationale Datenbanken;
2. das Warehouse selbst ist ein relationales DBMS;
3. Daten-Integration und -Extraktion erfolgen off-line, im allgemeinen im Batch-betrieb über Nacht, also zu einem Zeitpunkt, zu dem die operationale Datenbank vorübergehend abgeschaltet werden kann;
4. Quell-Datenbanken werden vollständig im Warehouse repliziert, d. h. eine Extraktion findet de facto nur auf rudimentärem Level statt;
5. das Warehouse wird in sog. Data Marts unterteilt, welche dann erst themenspezifische Untersuchungen, Auswertungen und OLAP erlauben.

Diese fünf Wartungsschritte für Data Warehouses sind in der Abbildung 18 dargestellt.

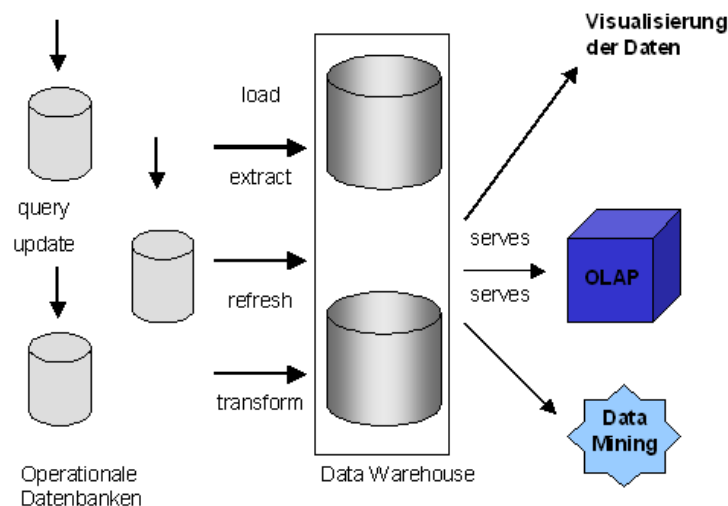


Abbildung 18: Data Warehouse Anwendungen<sup>188</sup>

Der Microsoft SQL Server 2005 zählt mit seinen Funktionalitäten zu den Business Intelligence-Plattformen der Gegenwart. Der neue SQL Server 2005 (Codename „Yukon“) soll Verbesserungen der Data Transformation Services enthalten. Die Data Transformation Services (DTS) beinhaltet eine ETL-Technologie (ETL = extrahieren, transformieren, laden). Im Vergleich zu den Vorgängermodellen wurde der SQL-Server für diese erweiterten Funktionen (z.B. grafische Fehlerbehebung, Erhöhung der Datenqualität, etc.) mit einer neu überarbeiteten Data-Transformation-Services (DTS) Architektur entwickelt.<sup>189</sup>

<sup>187</sup> Vossen (1999), S. 674

<sup>188</sup> Vossen (1999), S. 671

<sup>189</sup> vgl. o.V. (2003)

Der SQL Server dient mit seiner Basistechnologie vornehmlich als Datenspeicher. Andere Bestandteile wie z.B. die Analysis Services dienen der Datenauswertung. Hierbei geht es um die Entwicklung und Verwaltung der Daten.

### Technische Restriktionen des SQL Servers<sup>190</sup>

- 65535 Dimensionen pro Datenbank
- 65535 Ebenen pro Datenbank
- 128 Dimensionen pro Cube
- 256 Ebenen pro Cube
- 64 Ebenen pro Dimension
- 64000 untergeordnete Elemente je übergeordnetem Element

Die Architektur des SQL Servers mit seinen Analysekomponenten kann in verschiedene Schichten aufgeteilt werden.

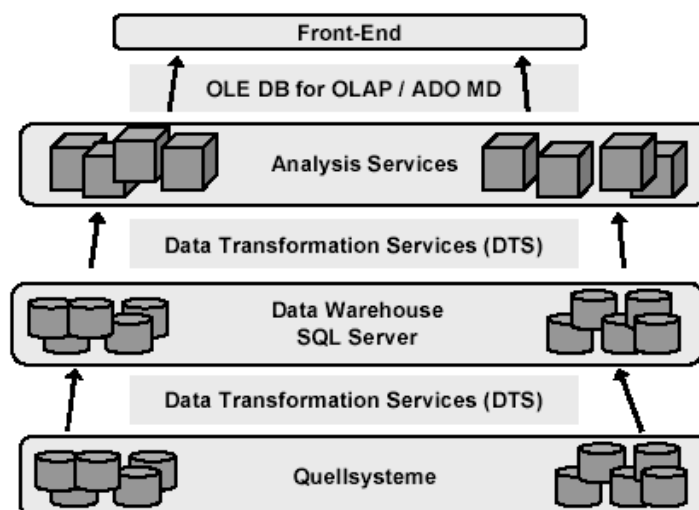


Abbildung 19: Schema-Architektur des Microsoft Data Warehouses<sup>191</sup>

Die Wesentlichen Erweiterungen des SQL Servers 2005 bestehen im Enterprise Data Management, Developer Productivity und Business Intelligence. Ein Betrieb ist sowohl im 32- als auch im 64-Bit-Modus möglich. Neu ist bei dieser Version auch, dass die Funktionalitäten von Enterprise Manager, Query Analyzer und Analysis Manager in einem Tool „SQL Server Management Studio“ subsumiert sind.<sup>192</sup>

### Technische Komponenten des Microsoft SQL Server 2005

Die Komponenten des Microsoft SQL Servers 2005 sind in der tabellarischen Übersicht mit der jeweiligen Funktion der Komponente aufgeführt.

<sup>190</sup> Hahne (2004b)

<sup>191</sup> Hahne (2004b)

<sup>192</sup> vgl. o.V. (2004)

Tabelle 8: Komponenten-Übersicht SQL Server 2005<sup>193</sup>

SQL Server 2005 - Komponente	Funktion
Data Transformation Services (DTS)	extrahieren, transformieren, laden
SQL Server 2005 relational database	Relationales Data Warehouse
SQL Server 2005 Analysis Services	Multidimensionales Datenbanksystem
SQL Server 2005 Reporting Services	Data mining
	geleitetes Berichtswesen
Microsoft Office products (Excel, Office Web Components, Data Analyzer, Sharepoint Portal)	Ad hoc Abfragen und Analysen
SQL Server 2005 Business Intelligence Development Studio	Entwicklungs-Tool für Datenbanksysteme
SQL Server 2005 SQL Server Management Studio	Management-Tool für Datenbanksysteme

Beim Erstellen einer Datenbank auf der SQL Server Plattform müssen zumindest die folgenden fünf Regeln eingehalten werden:<sup>194</sup>

- Dateien und Dateigruppen können nur von einer Datenbank verwendet werden. Dieser Punkt stellt hohe Anforderungen an die Unternehmensweite Datenmodellierung. Ansonsten müssen Dateien Redundant gespeichert werden.
- Eine Datei kann nur einer Dateigruppe angehören.
- Dateien und Transaktionsprotokolldaten müssen sich in verschiedenen Dateien befinden. Der Protokollspeicher wird immer unabhängig vom Datenspeicher verwaltet.  
Diese Anforderung wird im Microsoft SQL Server durch die separate Speicherung der *log-Datei* automatisch vorgenommen.
- Transaktionsprotokolldateien gehören nie einer Dateigruppe an.  
Dies ermöglicht eine bessere Übersicht, da Dateien getrennt gespeichert sein müssen. Andererseits werden dadurch mehr Dateien gebildet.
- Wenn eine Datei als Teil der Datenbank erstellt wurde, kann sie nicht in eine andere Dateigruppe verschoben werden. Wenn eine Datei verschoben werden soll, muss diese gelöscht und neu erstellt werden.

<sup>193</sup> in Anlehnung an Microsoft (2004a)

<sup>194</sup> vgl. Garcia et. al (2000), S. 168

Diese Anforderung stellt einen Nachteil des SQL Servers dar. Eine Tabelle zu kopieren und anschließend modifiziert erneut zu verwenden ist nicht möglich. Dies erhöht den Zeitaufwand der Datenbankerstellung.

**Store:** SQL Server 2005 blurs the lines between relational and multidimensional databases. You can store data in the relational database, in the multidimensional database, or use the new Proactive Cache feature to get the best of both worlds.<sup>195</sup>

### **Die neuen Datentypen des SQL Server 2005<sup>196</sup>**

Es wurden ein paar neue Datentypen entwickelt, welche gerade für den Bereich Data Warehouse nützlich sein werden. Unter den Datentypen varchar(max), nvarchar(max) und varbinary(max) können Daten bis zu einem Speichervolumen von 2 GB gespeichert werden. Diese Datentypen sind für die Speicherung der erweiterten Metadaten und anderen Beschreibungsinformationen in einem Data Warehouse geeignet. Dadurch ergeben sich geeignete Alternativen zu den bekannten Datentypen **text**, **ntext** und **image**.

### **Fazit**

Mit dem SQL Server 2005 bietet Microsoft eine neue Version des SQL-Servers an. Diese Version bietet einige Erweiterungen und Neuerungen, welche teilweise bereits erläutert wurden oder umfolgenden Text noch dargestellt werden. Durch die zahlreichen Erweiterungen der BI-Tools werden die Data Warehouse Funktionalitäten weiter optimiert.

## **4.2 Business Intelligence mit SQL Server 2005 – Analysis Services**

Die Nutzung von unternehmensweit verfügbarem Wissen wird heute unter dem Namen *Business Intelligence* zusammengefasst und bildet somit das Frontend des Data Warehouse.<sup>197</sup>

Beim Microsoft SQL Server handelt es sich um ein sogenanntes physikalisches Data Warehouse. Der SQL Server kennt die ROLAP, MOLAP und HOLAP Anwendungen.

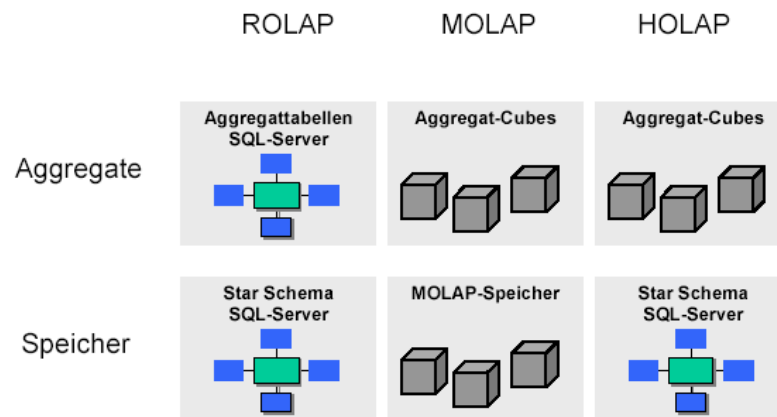
Für den Zugriff auf einen mehrdimensionalen Cube kann Excel verwendet werden.

---

<sup>195</sup> Microsoft (2004a)

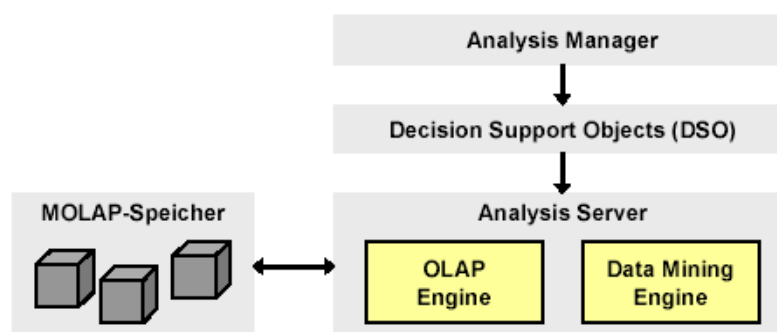
<sup>196</sup> vgl. Microsoft (2004a)

<sup>197</sup> Lehmann (2001), S. 47

Abbildung 20: SQL Server Speichervarianten<sup>198</sup>

#### 4.2.1 Analysis Services Enhancements

Bei Analysis Services handelt es sich um eine Sammlung von Tools, durch welche die analytische Daten-Verarbeitung (Entwicklung und Verwaltung der Daten) unterstützt wird. In diesen Tools ist der Analysis-Server integriert. Gespeicherte multidimensionale Datencubes werden innerhalb dieses Servers gebildet, verwaltet und gespeichert.<sup>199</sup>

Abbildung 21: Architektur Analysis Services<sup>200</sup>

Die Analysis Services setzen das Vorhandensein eines relationalen Data Warehouse voraus, welches durch die Faktentabelle charakterisiert ist. Diese Faktentabelle beinhaltet die Detailwerte für die Measures. Die Informationen, welche zur Berechnung von Aggregatwerten benötigt werden, sind in den Dimensionstabellen gespeichert.

Analysis Services unterstützt die OLAP-Verarbeitung. Durch diese Verarbeitung greifen die Analysis Services auf die im Data Warehouse gespeicherten Daten zu, um damit Auswertungsberechnungen vorzunehmen.<sup>201</sup>

<sup>198</sup> Hahne (2004b)

<sup>199</sup> vgl. Garcia et.al (2000), S. 706

<sup>200</sup> Hahne (2004b)

<sup>201</sup> vgl. Garcia et.al (2000), S. 705



Die durch Analysis Services ausgewerteten Dimensionsinformationen werden in jeweils eigenen Datenstrukturen abgespeichert. Aus diesem Grund ergibt beim Microsoft SQL Server das Starschemadesign gegenüber dem Snowflake-Schemadesign keine Geschwindigkeitsvorteile.<sup>202</sup>

Die Analysis Services 2005 bestehen im wesentlichen aus den OLAP und Data Mining Funktionalitäten.<sup>203</sup>

Die Analysis Services fügen sich zwischen die relationale Datenbank und die Berichtsanwendung ein und funktionieren als Zwischenschicht, in der die Daten aus dem Data Warehouse in eine Form umgewandelt werden, die für schnelle und flexible Analyseberichte geeignet ist.<sup>204</sup>

Die primäre Komponente der Analysis Services ist der Analysis-Server, der als Dienst unter Microsoft Windows ausgeführt wird. Der Analysis-Server extrahiert die Informationen, aus dem Data Warehouse und bildet daraus einen sogenannten *Cube*, einen Würfel.<sup>205</sup>

Um die Cubes zu definieren und mit den Daten zu füllen wird der Analysis-Manager verwendet. Der Analysis-Manager kommuniziert mit der *Decision Support Objects (DSO)* Schicht (siehe Abbildung 22) . Bei der DSO handelt es sich um einen „Satz von Funktionen“. Mit diesen Funktionen kann eine Anwendung die Analysis Services steuern.<sup>206</sup>

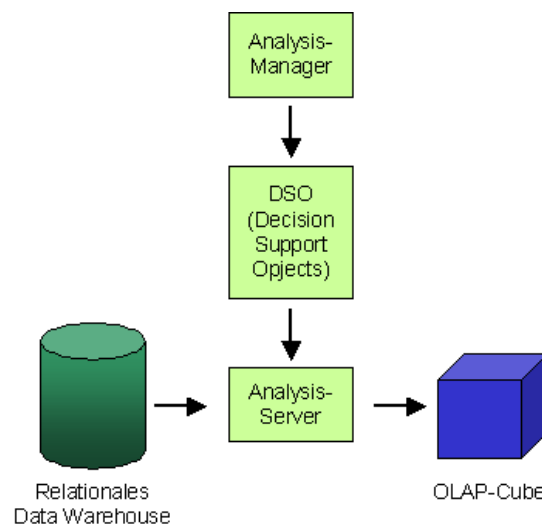


Abbildung 22: Der administrative Zugriff auf den Analysis-Server<sup>207</sup>

<sup>202</sup> vgl. Jacobson (2000), S. 35

<sup>203</sup> vgl. Microsoft (2004a)

<sup>204</sup> Jacobson (2000), S. 36f.

<sup>205</sup> Jacobson (2000), S. 37

<sup>206</sup> vgl. Jacobson (2000), S. 39

<sup>207</sup> Jacobson (2000), S. 40

In der gleichen Weise, wie ein Administrationsprogramm (etwa der Analysis-Manager) DSO zur Kommunikation mit dem Analysis-Server nutzt, so nutzt eine Clientanwendung (etwa Excel) *OLE DB für OLAP* zur Kommunikation mit dem PivotTable Service.<sup>208</sup>

Die Visualisierung der Daten in einer grafischen Ansicht erfolgt in Analysis Services im Normalfall über die Auswertung von mehrdimensionalen Cubes. Die einzelnen Cubes können über das Star-Schema oder Snowflake-Schema erstellt werden. Auf den Inhalt von Cubes, sowie das Star- und Snowflake-Schema ist bereits ausführlich eingegangen worden.

### **Aggregation von Daten**

Analysis Services erstellt in der Faktentabelle Spaltenaggregat, die auf den Daten in den Dimensionstabellen basieren. So können beispielsweise Dimensionstabellen zu Personalfragen verwendet werden, um die Verkaufsdaten nach Mitarbeiter geordnet zu summieren. Dimensionstabellen zu Elementen können zur Aufsummierung von Daten bezüglich der Elemente verwendet werden. Da die Aggregationen anhand der Dimensionstabellen unterschiedlich große Slices oder Dimensionen der Daten ergeben, wird ein virtueller Datencube erstellt.<sup>209</sup>

Mithilfe des Analysis Services-Assistenten können Aggregationen erstellt werden, die dann zur Erstellung von Geschäftsmodellen oder bei wichtigen Unternehmensentscheidungen verwendet werden.<sup>210</sup>

Analysis Services wird dadurch mit einer SQL Server-Datenbank verbunden, dass für den Server eine ODBC-Systemdatenquelle (Open Database Connectivity) eingerichtet wird.<sup>211</sup>

### **Semantikverlust**

Die in dieser Arbeit beschriebenen Datenmodelle eignen sich dazu, mehrdimensionale Daten in einem DBMS multidimensional zu speichern. Bei der Übertragung in ein relationales Datenbanksystem besteht aber die Gefahr nicht alle semantischen Informationen der multidimensionalen Daten in die Relationen zu übertragen.

Das multidimensionale Datenmodell unterscheidet zwischen Dimensionen und Kenngrößen. Dieses Kriterium geht bei der relationalen Speicherung in einer Faktentabelle explizit verloren. Die Differenzierung kann zwar durch das Fremdschlüsselmerkmal des Attributs zurückverfolgt werden, die direkte Unterscheidung kann jedoch nicht erfolgen. Auch der Aufbau der einzelnen Dimensionen erfolgt bei der Transformation nicht verlustfrei. Sowohl die Drill-Pfade, als auch die Stufen der Klassifizierungen sind nicht vollständig übernehmbar.<sup>212</sup>

---

<sup>208</sup> Jacobson (2000), S. 41

<sup>209</sup> Garcia et. al (2000), S. 709

<sup>210</sup> Garcia et. al (2000), S. 709

<sup>211</sup> Garcia et. al (2000), S. 714

<sup>212</sup> vgl. Bauer, Günzel (2001), S. 207

### 4.2.2 Reporting Services

Die Version des SQL Server 2005 wird von Microsoft durch die neue Komponente „Reporting Services“ für die Business Intelligence (BI) Anwendung erweitert. Die Funktionalitäten der Reporting Services sollen, der BI-Vision entsprechend, die richtigen Informationen an die richtigen Menschen innerhalb des Unternehmens an jeden beliebigen Ort zu liefern.

Die Reporting Services ist eine serverbasierte Plattform, um Reports auf Basis der Unternehmensdaten zu erstellen, zu bearbeiten und interaktive Reports zu ermöglichen. Über die ausgeweitete Application Programming Interface (API) der „Routing Services“ wird den Software-Entwicklern ermöglicht, weitere Reportquellen in die Reporting Services einzubinden.<sup>213</sup>

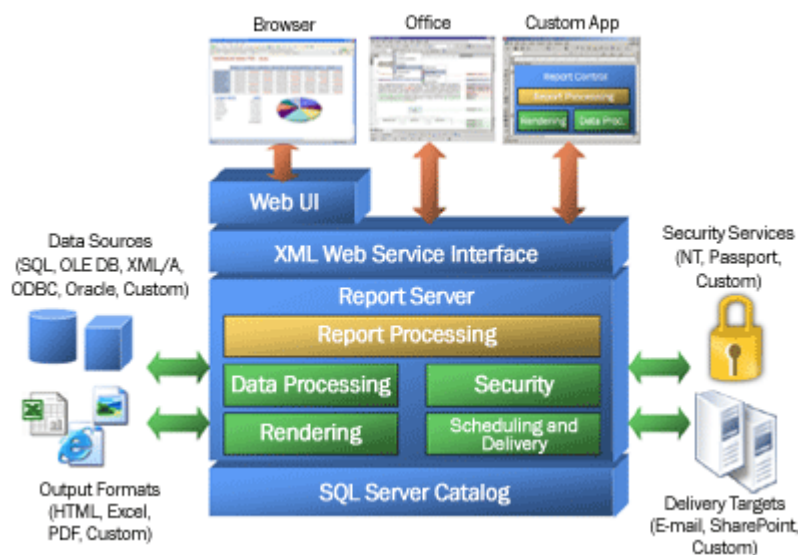


Abbildung 23: Architektur der Reporting Services – SQL Server 2005<sup>214</sup>

Reporting Services nutzen die Anwendungen des SQL Server 2005. Dabei beinhalten die Reporting Services:<sup>215</sup>

- Werkzeug (tools) um Reports zu erstellen, zu bearbeiten und zu visualisieren
- Anwendungen um die Reports zu speichern und zu bearbeiten
- Eine erweiterbare Architektur und APIs um die erstellten Reports an andere IT-Umgebungen exportieren zu können, bzw. andere Anwendungen in die Architektur zu integrieren.

<sup>213</sup> vgl. Nunn (2004)

<sup>214</sup> Microsoft (2004b)

<sup>215</sup> vgl. Nunn (2004)

### 4.3 Zusammenfassung

Die BI-Plattform des SQL Servers 2005 von Microsoft kann zusammengefasst charakterisiert werden:<sup>216</sup>

- Bei großen und komplexen Data Warehouses auf einfache Weise Abfragen zu erstellen
- Niedrige Unterhaltskosten
- Einfache Report- und Analysesysteme um die Anwendungen der kleineren Unternehmen oder einzelne Abteilungen größerer Unternehmen einfach darstellen zu können
- Bereitstellung analytischer Informationen
- Closed-Loop-Analysen und Data Mining
- Eingebundene Systeme zur Erweiterung der BI-Erreichbarkeit

Die einzelnen Tools, *SQL Server relational database*, *DTS*, *Reporting Services* und *Analysis Services OLAP und Data Mining* beinhalten neue Funktionalitäten, welche die BI-Plattform von Microsoft erweitert. Dabei hat Microsoft versucht, die Anwendungen möglichst einfach zu gestalten, damit unter vertretbarem Aufwand geeignete Reports erstellbar sind.

---

<sup>216</sup> vgl. Microsoft (2004a)

## 5 Sales Fallstudie

In der Fallstudie werden die bereits dargestellten Schemata auf „Yukon“ übertragen. Der Fallstudie liegt fiktives Musterunternehmen zu Grunde, welches im folgenden kurz dargestellt werden soll.

IBI Technologies ist ein Handelsunternehmen für Computer. Die Produktkategorien sind in:

- Desktops
- Notebooks
- Tablet PCs und
- Pocket PCs

eingeteilt.

Als Hersteller bzw. Lieferanten wird IBI Technologies Verbindungen von:

- Acer
- Apple
- Compaq
- HP
- IBM und
- Sony

beliefert.

Die IBI Technologies ist in die Regionen Europa (Sitz der Zentrale), Nordamerika, Südamerika, Afrika, Asien-Pazifik und Australien aufgeteilt.

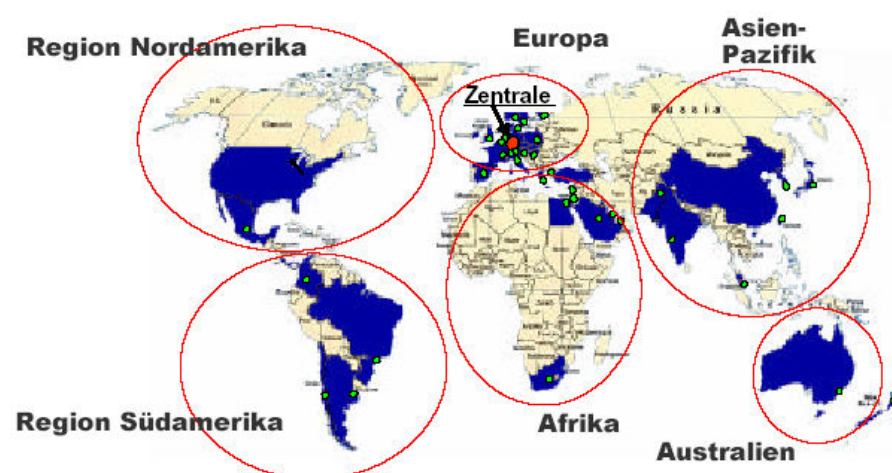


Abbildung 24: Regionen-Aufteilung der Fallstudie

Es gilt verschiedene wirtschaftliche Analysen im Bereich Vertrieb zu untersuchen. Folgende Grundfragen werden als Überlegungen angestellt.

**Bewertung der Performance der Vertriebseinheiten:**

- Welche **Filiale** war im vergangenen **Monat** am erfolgreichsten? (Umsatz-/ Ergebnisentwicklung)
- Welches sind die **Produkte**, die am meisten verkauft werden?
- Welches sind die Produkte, die den besten Gewinn machen?
- Gibt es gravierende Abweichungen der Umsätze vom geplanten Soll? (Wenn ja, woher resultieren sie?)

In der Vertriebsorganisation werden die Kennzahlen von Verkäufen rund um die Filialen, Produkte und die Zeit betrachtet.

**Betriebswirtschaftliche Analysen im Bereich Marketing – Bewertung der Wirtschaftlichkeit der Werbekampagnen:**

- Welche Kampagnen wurden im vergangenen Monat mit welcher Filiale mit welcher Werbeart durchgeführt?
- Welche Werbearten wurden am meisten eingesetzt?
- Welchen Anteil hatte die Werbeart „TV/Rundfunk“?
- Was hat eine Werbekampagne insgesamt gekostet?

Das Marketing betrachtet die Kennzahlen von Kampagnen rund um die Filialen, Werbearten und die Zeit.

**5.1 Datenmodelle der Fallstudie**

Aus diesen Informationen über die IBI Technologies kann nach dem oben vorgestellten Vorgehensmodell ein ER-Modell erstellt werden. In den folgenden Abbildungen werden die Modelle für den Vertrieb und das Marketing dargestellt.

Das Entity-Relationship-Modell, das ME/R-Modell und das Star-Schema gehören zu der Gruppe der konzeptionellen Datenmodelle. Neben diesen drei Modellen ist dazu noch das Dimensional Fact Modeling oder das Kennzahlendatenmodell hinzu zu zählen.<sup>217</sup>

Zunächst erfolgt die Bestimmung der Entitäten. Anschließend werden die Beziehungen bestimmt. In der Fallstudie sind dies die Beziehungen „*wirbt*“ und „*verkauft*“.

Die Entitäten müssen für die Dokumentation kurz beschrieben werden.

Tabelle 9: Beschreibung der Entitäten für IBI Technologies

Entitätstyp	Beschreibung
PRODUKT	Beschreibt die hergestellte Leistung
FILIALE	Bezeichnet den Verkaufspunkt der Produkte

<sup>217</sup> Hahne (2004b)

WERBEART	Enthält alle Informationen der Werbemittel
ZEIT	Stellt die Aspekte der Zeit dar
VERKAUF (schwache Entität)	Daten der Umsätze in Abhängigkeit von Produkt, Filiale und Zeit
KAMPAGNE (schwache Entität)	Daten der Werbeart in Abhängigkeit von Filiale, Zeit und Werbeart

**Starker Entitätstyp:** Ein Entitätstyp, dessen Existenz nicht von einem anderen Entitätstyp abhängt.<sup>218</sup>

**Schwacher Entitätstyp:** Ein Entitätstyp, dessen Existenz von einem anderen Entitätstyp abhängt.<sup>219</sup>

In einem der ersten Schritte werden alle Entitäten und Relationen identifiziert. Falls bereits möglich, werden schwache und starke Entitäten bereits festgelegt.



Abbildung 25: Identifizierte Entitäten der Fallstudie

Anschließend wird das Entity-Relationship-Modell für den Vertrieb und das Marketing aufgebaut. Dabei werden die definierten Entitäten für den Geschäftsvorfall zueinander in Beziehung gesetzt.

<sup>218</sup> Connolly, Begg, Strachan (2002), S. 186

<sup>219</sup> Connolly, Begg, Strachan (2002), S. 186

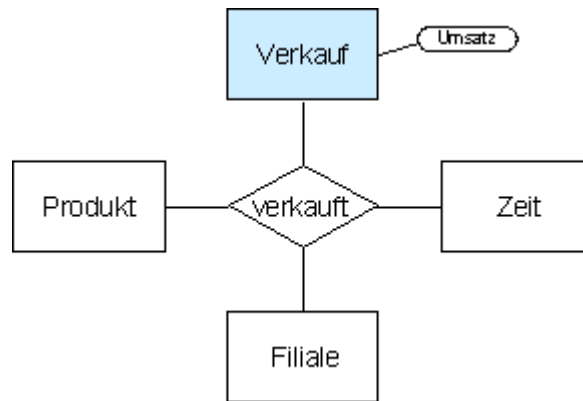


Abbildung 26: ER-Modell der Fallstudie – Vertrieb

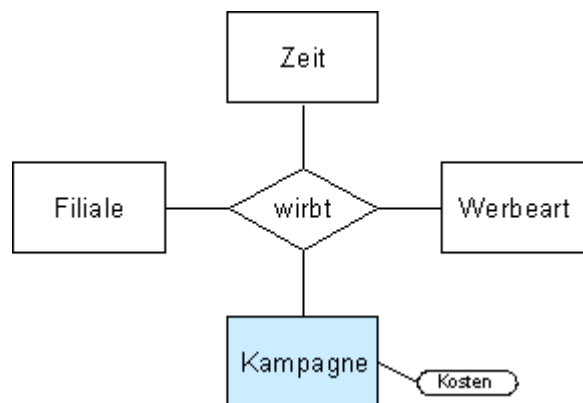


Abbildung 27: ER-Modell der Fallstudie – Marketing

Da es sich beim Vertrieb und Marketing um zwei Geschäftsvorfälle handelt, welche miteinander in Verbindung stehen und die Entitäten Filiale und Zeit in beiden betrachteten Geschäftsvorfällen eingebunden sind, werden diese zu einem Business-Szenario zusammengefasst. Dadurch entsteht das in Abbildung 28 dargestellte Business-Szenario. Dies ist die ER/M-Darstellung der Fallstudie.



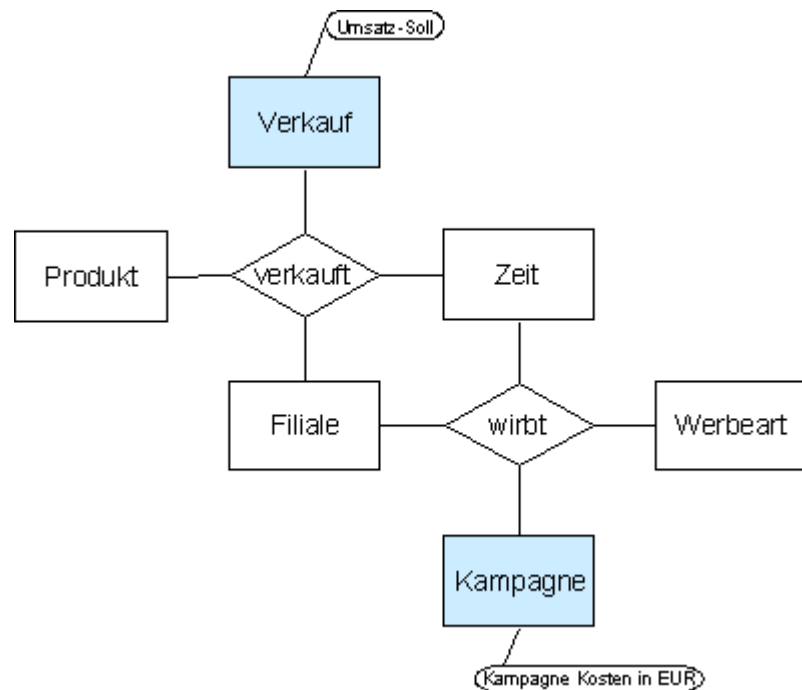


Abbildung 28: Business-Szenario der Fallstudie im Datenmodell

Um die Entitäten näher zu spezifizieren werden ihnen die beschreibenden Attribute zugeordnet. Das Schlüsselattribut wird durch die Unterstreichung kenntlich gemacht.

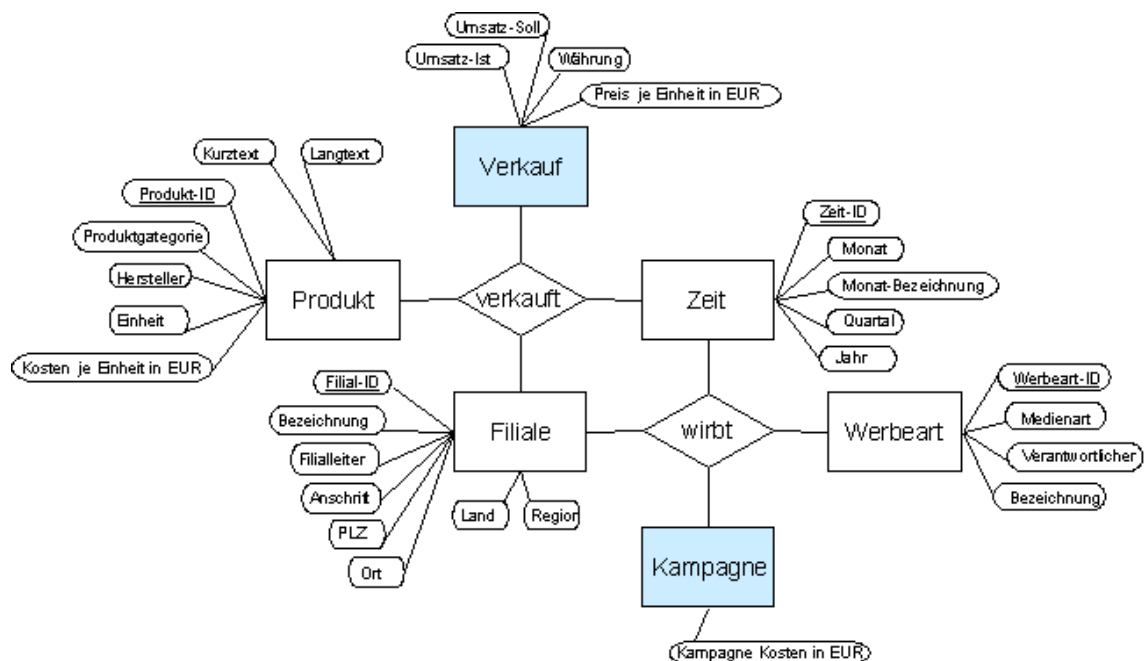


Abbildung 29: Business-Szenario im Datenmodell mit Attributen

Nachdem alle Entitäten mit den Attributen identifiziert wurden müssen die Datentypen der Felder festgelegt werden. Für das Beispiel der Fallstudie werden die benötigten Datentypen aufgeführt und charakterisiert.

Tabelle 10: Datentypen und deren Beschreibungen<sup>220</sup>

Datentyp	Name des Datentyps	Beschreibung
int	Integer	Ganze Zahl, üblicherweise vier Byte; unerlässlich
char (n)	Character (n)	Zeichenketten mit genau n Zeichen bis 255 Zeichen
varchar (n)	Character varying	Zeichenketten mit höchstens n Zeichen bis 255 Zeichen
text	Text	Beschreibungen bis 1.073.741.823 Zeichen
numeric (p)	Numeric (p)	Zahl mit mindestens p Stellen
decimal (p, q)	Decimal (p, q)	Dezimalzahlen mit mindestens p Stellen, davon q hinter dem Dezimalpunkt

Die hier definierten Datentypen werden schließlich verwendet. Daraus ergibt sich die in Tabelle 11 ersichtliche Datenstruktur für die einzelnen Attribute. Außerdem wird bei diesem Schritt auch festgelegt, welche Felder mit einem Datenwert belegt sein müssen. Diese sind in der letzten Spalte der folgenden Tabelle mit einem >>Nein<< belegt. Dies besagt, dass die entsprechenden Felder in der Datenbank mit einem Wert belegt werden müssen. Das DBMS erlaubt in diesem Zusammenhang nicht, das Feld ohne Wert besetzt zu lassen. Es erfolgt eine zwingende Eingabeaufforderung.

Tabelle 11: Entitäten mit den Attributen und deren Beschreibung

Entitätstyp	Name der Attribute	Beschreibung	Datentyp	NULL Ja / Nein
PRODUKT	<u>Produkt-ID</u>	Eindeutige Bezeichnung des Produkts	int	Nein
	Produktkategorie	Produktgruppe der hergestellten Leistung	varchar (25)	Ja
	Hersteller	Bezeichnung des Produktionslabels	varchar (25)	Ja
	Einheit	Maßeinheit der Produktmenge	varchar (5)	Ja
	Kosten je Einheit in EUR	Kosten in EUR pro Einheit der Produktmenge	decimal (20, 2)	Ja
	Kurztext	Kurzbezeichnung (Typ)	text	Ja
	Produkt-Text	Produktbeschreibung	text	Ja

<sup>220</sup> in Anlehnung an Matthiessen, Unterstein (2000), S. 143ff. sowie Microsoft (2004c)

FILIALE	<u>Filial-ID</u>	Eindeutige Bezeichnung der Filiale	int	Nein
	Bezeichnung	Name der Filiale	text	Ja
	Filialleiter	Name des Filialleiters	varchar (25)	Ja
	Anschrift	Strassenname der Filiale	text	Ja
	PLZ	Postleitzahl der Anschrift	varchar (25)	Ja
	Ort	Name des Orts der Filiale	text	Ja
	Land-ID	Eindeutige Bezeichnung des Lands	char (10)	Ja
	Sprache	Sprache des Lands	varchar (10)	Ja
	Beschreibung	Name des Lands	varchar (50)	Ja
	Region-ID	Eindeutige Bezeichnung der Region	char (10)	Ja
	Sprache	Sprache der Region	varchar (10)	Ja
	Text	Name der Region	varchar (50)	Ja
WERBEART	<u>Werbeart-ID</u>	Eindeutige Bezeichnung der Werbeart	int	Nein
	Medienart	Werbemedium	text	Ja
	Verantwortlicher	Ansprechpartner der Werbung	varchar (25)	Ja
	Bezeichnung	Beschreibung der Werbung	text	Ja
ZEIT	<u>Zeit-ID</u>	Eindeutige Bezeichnung der Zeit	int	Nein
	Monat	Numerische Bezeichnung von Monat und Jahr	varchar (10)	Ja
	Bezeichnung	Name des Monats (textuell)	varchar (25)	Ja
	Quartal	Quartalseinheit	varchar (10)	Ja
	Jahr	Jahresbezeichnung (numerisch)	numeric (4)	Ja
VERKAUF	Umsatz-Ist	Getätigter Umsatz	decimal (20, 2)	Ja
	Umsatz-Soll	Geplanter Umsatz	decimal (20, 2)	Ja
	Währung	Einheit der Währung	varchar (5)	Ja
	Preis je Einheit in EUR	Kosten je Produkt-Einheit, in EUR dargestellt	decimal (20, 2)	Ja
KAMPAGNE	Kampagnenkosten in EUR	Kosten der jeweiligen Werbekampagne in EUR	decimal (20, 2)	Ja

Anschließend werden die ermittelten ER-Modelle in die Star-Schemata übertragen.

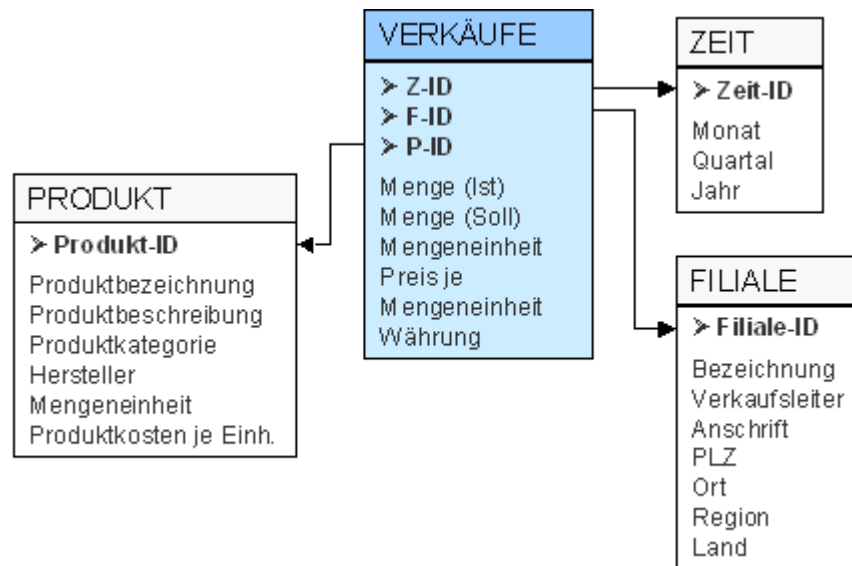


Abbildung 30: Star-Schema Vertrieb

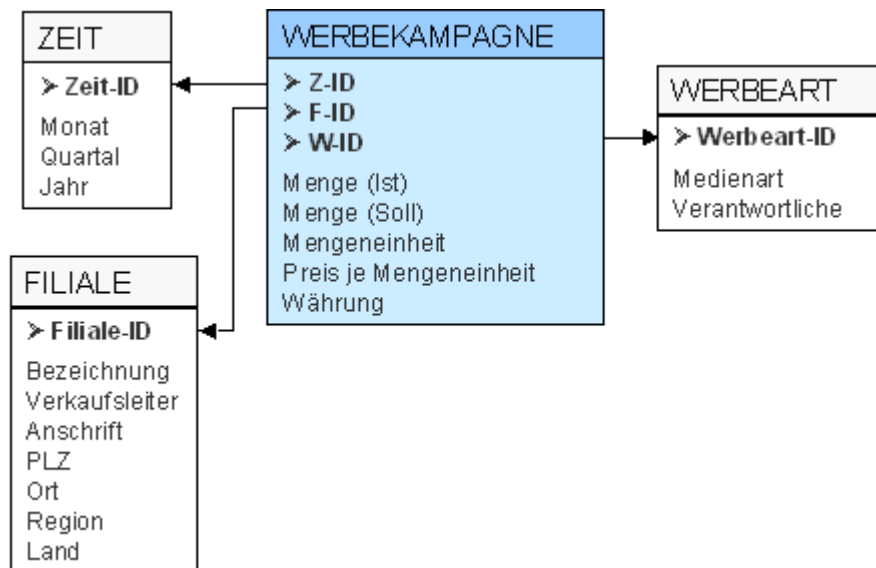


Abbildung 31: Star-Schema Marketing

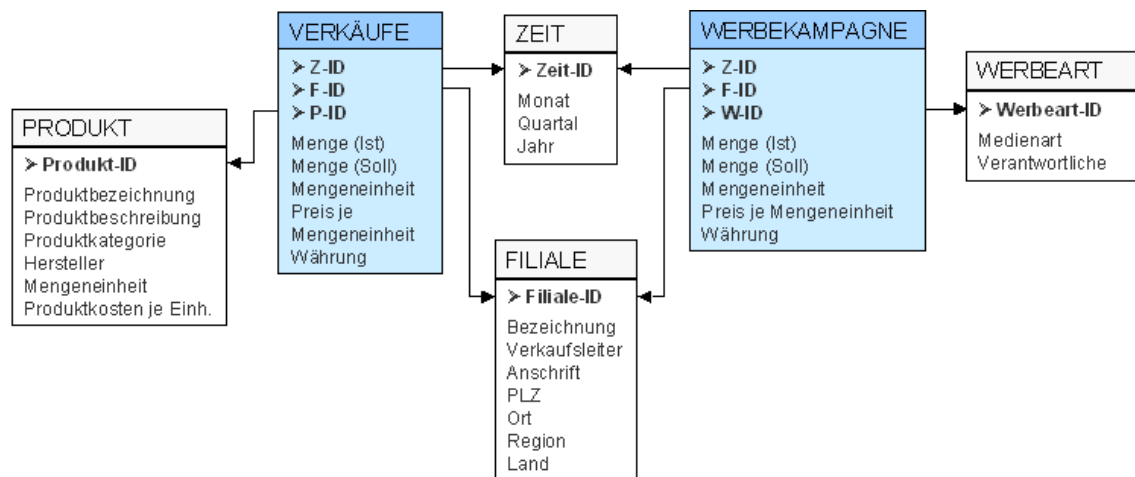


Abbildung 32: Business-Szenario im Galaxy-Schema

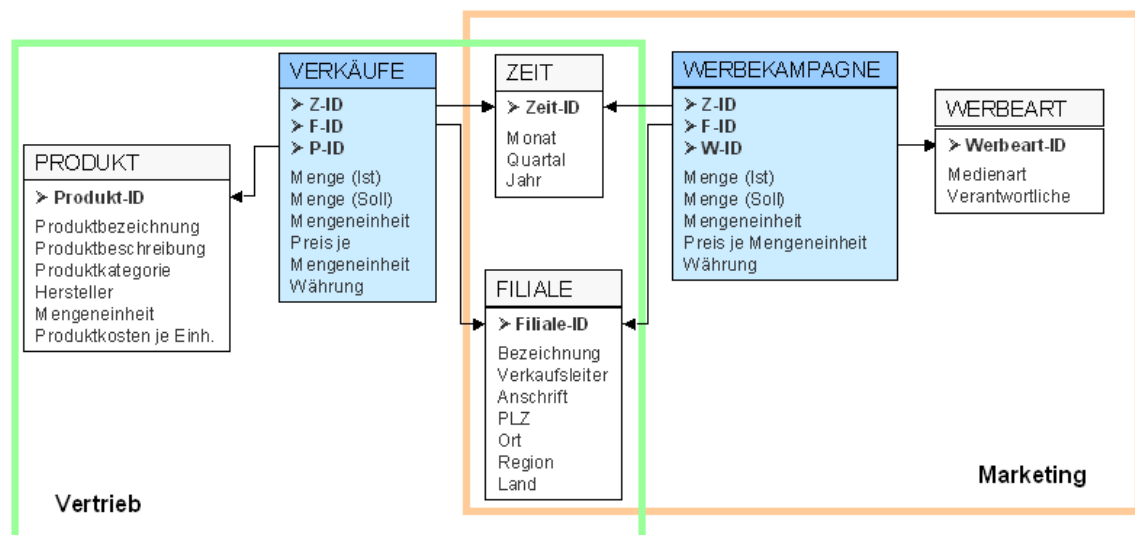


Abbildung 33: Business-Szenario im Galaxy-Schema nach Vertrieb und Marketing

In der nächsten Abbildung werden die Star-Schemata analog als Cubes betrachtet und im Modell vereint. Dadurch entsteht der Multi-Cube, welcher hier als „Sales“ bezeichnet wurde. Dieser stellt das Business-Szenario in der mehrdimensionalen Struktur, als Würfel dargestellt, dar.

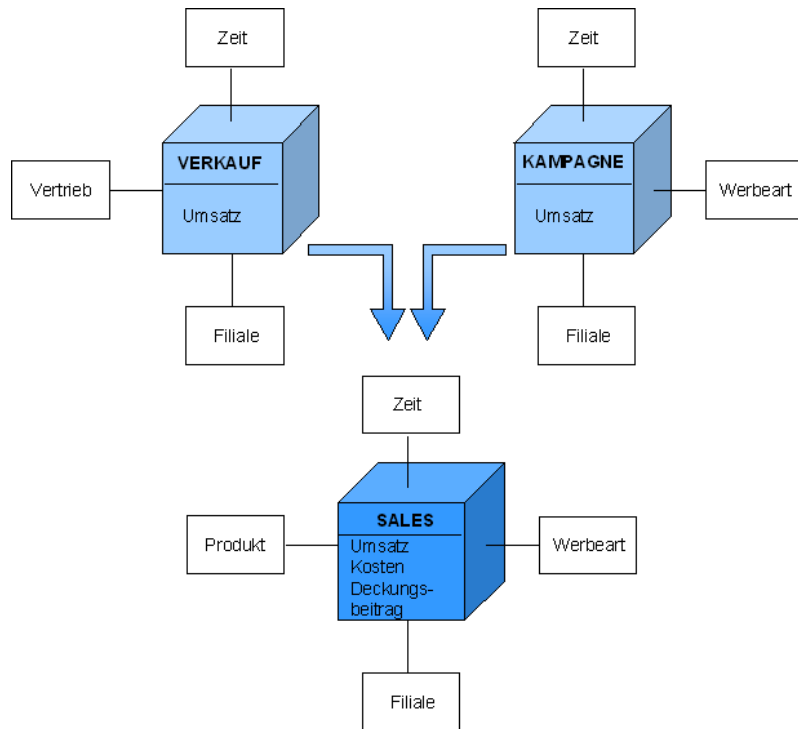


Abbildung 34: Business-Szenario als mehrdimensionales Datenmodell

Durch diese Konsolidierung der Cubes *Verkauf* und *Kampagne* entsteht der virtuelle Cube **Sales**.

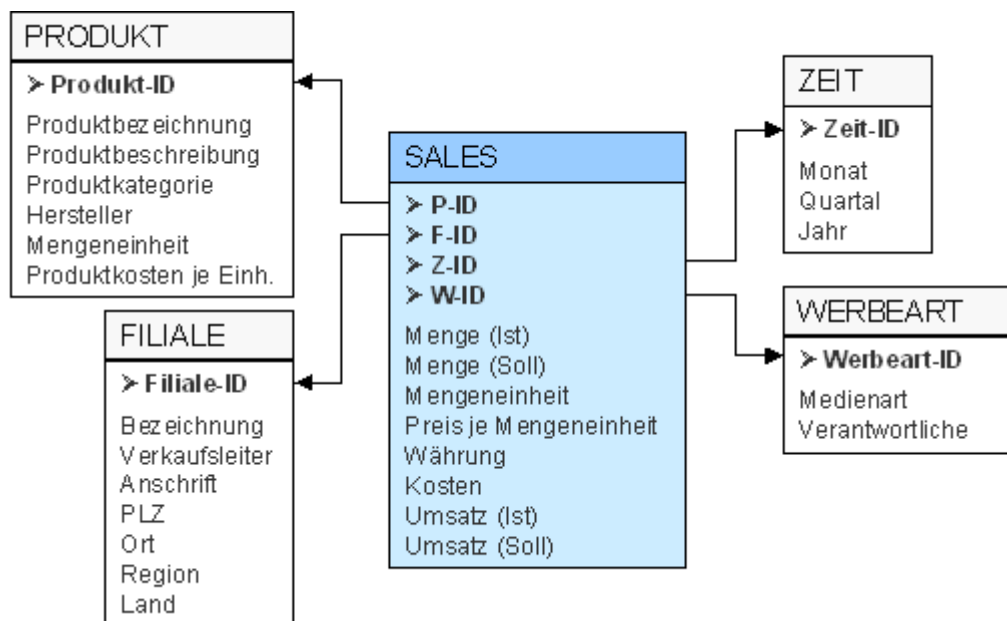


Abbildung 35: Sales-Fallstudie im Star-Schema

Als letzter Schritt vor der Implementierung SQL Server Datenbank werden die beiden Dimensionstabellen aus Gründen der Übersicht denormalisiert. Wie bereits beschrieben, entsteht somit ein Partial Snowflake-Schema.

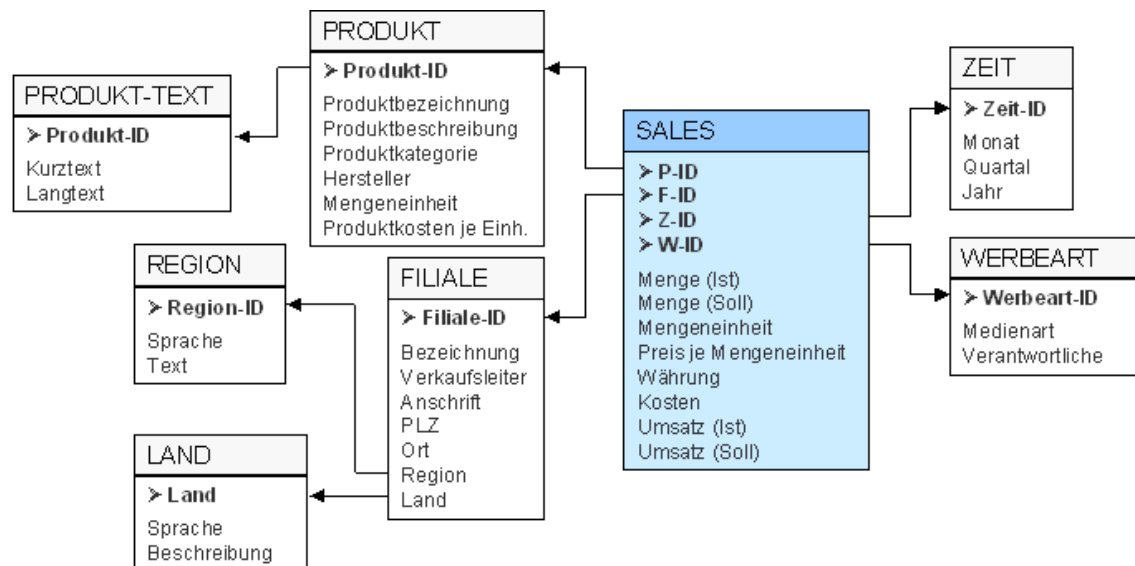


Abbildung 36: Sales-Fallstudie im Partial Snow-Flake Schema

## 5.2 Transformation der Datenmodelle in das Datenbankmanagementsystem

Das implementierte Datenmodell besitzt dann diese Form in SQL Server 2005. Aus dieser Ansicht werden die Tabellen und deren Beziehungen ersichtlich. Um eine Abfrage zu erstellen werden die zu benötigenden Tabellen ausgewählt und im Data Explorer angezeigt.

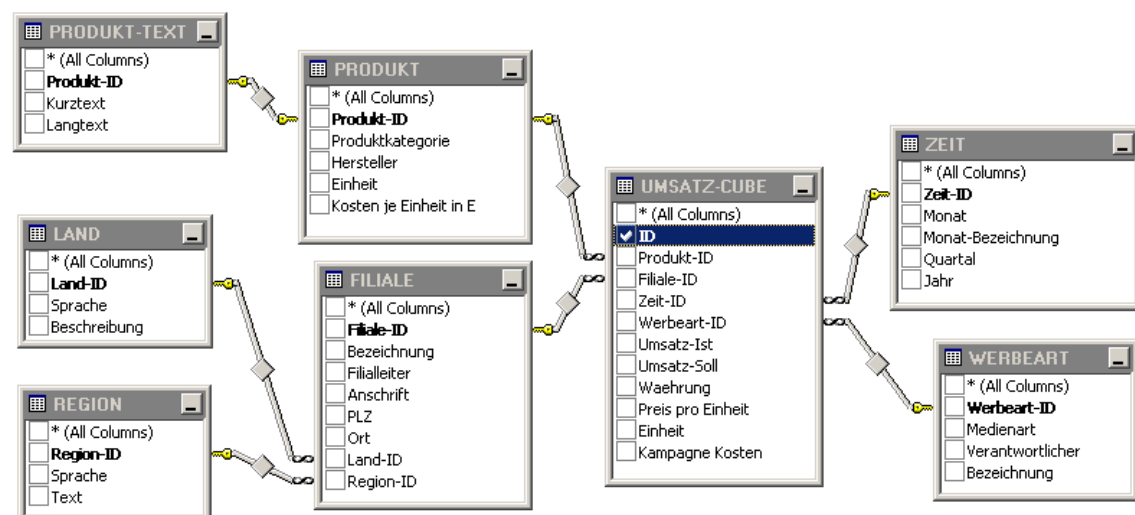


Abbildung 37: Relationenübersicht mit den ermittelten Beziehungen

In der Abbildung 38 ist eine Beispielabfrage und ein Ausschnitt der Ergebnismenge aufgeführt.

	Column	Alias	Table	Output	Sort Type	Sort Order	Filter	Or...	
	Bezeichnung		FILIALE	<input checked="" type="checkbox"/>					
	Filialeleiter		FILIALE	<input checked="" type="checkbox"/>					
	Produktkategorie		PRODUKT	<input checked="" type="checkbox"/>					
▶	Hersteller		PRODUKT	<input type="checkbox"/>				= 'Acer'	
	Langtext		[PRODUKT-...	<input checked="" type="checkbox"/>					
	Text		REGION	<input checked="" type="checkbox"/>					
	Monat		ZEIT	<input checked="" type="checkbox"/>				= '12.2003'	

```

SELECT  dbo.FILIALE.Bezeichnung, dbo.FILIALE.Filialeleiter, dbo.PRODUKT.Produktkategorie, dbo.[PRODUKT-TEXT].Langtext, dbo.REGION.Text,
        dbo.ZEIT.Monat
FROM    dbo.REGION INNER JOIN
        dbo.FILIALE INNER JOIN
        dbo.LAND ON dbo.FILIALE.[Land-ID] = dbo.LAND.[Land-ID] ON dbo.REGION.[Region-ID] = dbo.FILIALE.[Region-ID] INNER JOIN
        dbo.[UMSATZ-CUBE] ON dbo.FILIALE.[Filiale-ID] = dbo.[UMSATZ-CUBE].[Filiale-ID] INNER JOIN
        dbo.[PRODUKT-TEXT] INNER JOIN
        dbo.PRODUKT ON dbo.[PRODUKT-TEXT].[Produkt-ID] = dbo.PRODUKT.[Produkt-ID] ON
        dbo.[UMSATZ-CUBE].[Produkt-ID] = dbo.PRODUKT.[Produkt-ID] INNER JOIN
        dbo.ZEIT ON dbo.[UMSATZ-CUBE].[Zeit-ID] = dbo.ZEIT.[Zeit-ID]
WHERE   (dbo.PRODUKT.Hersteller = 'Acer') AND (dbo.ZEIT.Monat = '12.2003')

```

	Bezeichnung	Filialeleiter	Produktkategorie	Langtext	Text	Monat
▶	Global Computer ...	Yvonne Moncada	Tablet PC	900-MHz PIII-M ...	Südamerika	12.2003
	Global Computer ...	Yvonne Moncada	Notebook	20-GHz Celeron...	Südamerika	12.2003
	Global Computer ...	Yvonne Moncada	Notebook	180-GHz Pentiu...	Südamerika	12.2003
	Global Computer ...	Yvonne Moncada	Notebook	20-GHz Celeron...	Südamerika	12.2003
	Global Computer ...	Yvonne Moncada	Notebook	22-GHz Celeron...	Südamerika	12.2003
	Global Computer ...	Yvonne Moncada	Notebook	160-GHz Pentiu...	Südamerika	12.2003
	Global Computer ...	Roland Mendel	Tablet PC	900-MHz PIII-M ...	Europa	12.2003
	Global Computer ...	Roland Mendel	Notebook	20-GHz Celeron...	Europa	12.2003
	Global Computer ...	Roland Mendel	Notebook	180-GHz Pentiu...	Europa	12.2003

1 of 306

Abbildung 38: Beispielabfrage über mehrere Dimensionen der Fallstudie



## 6 Zusammenfassung und Ausblick

In der Informationstechnologie erhält die Versorgung von Fach- und Führungskräften mit adäquaten analyserelevanten Informationen einen neuen Stellenwert, da sich derzeit ein neuer Wandel von einer operativen zu einer analyseorientierten und strategischen Ausrichtung der Informationsverarbeitung vollzieht.<sup>221</sup>

Die Informationsversorgung für strategische Entscheidungen erfolgt nach heutigem Wissensstand am geeignetsten aus Daten mehrdimensionaler Datenbankmanagementsysteme. Diese Speicherstruktur deckt den Aufwand – Nutzen in zweckentsprechender Weise ab. Das Know-How aus der relationalen Datenbanktechnik kann dabei weitgehend verwendet werden.

Die spezifische Datenbank-Technologie basiert auf dem Relationenmodell. Die spezielle Form zur Abbildung mehrdimensionaler Datenstrukturen in diesem Modell ist unter dem Namen Star-Schema eingeführt und mittlerweile in einer Vielzahl von Ausprägungen verfügbar.<sup>222</sup>

Diese Modelle wurden in der Arbeit ausführlich behandelt. In der Literaturliste befinden sich zahlreiche Literaturangaben zur Vertiefung der Thematik.

Auf der Ebene der semantischen Modellierung fehlt ein leicht handhabbares, dem Mitarbeiter in der Fachabteilung verständliches, Modell insbesondere zur Berücksichtigung betriebswirtschaftlicher Fragestellungen. Mit dem Star-Schema und seinen Varianten stehen für die logische Modellierung zum Einsatz relationaler Datenbanksysteme als Data Warehouse-Datenbank mittlerweile anerkannte Modelle zur Verfügung. Mit einheitlichen logischen Modellen für den Einsatz multidimensionaler Datenbanksysteme liegt es dagegen im Argen.<sup>223</sup>

Für mittelständische Unternehmen wird es in Zukunft wichtiger die Bereitstellung qualitativer Daten zur Entscheidungsunterstützung auswerten zu können. Dadurch werden relevante Zusammenhänge erkennbar und die Kommunikation zur Entscheidung liegt auf einer gemeinschaftlichen Informationsbasis begründet.

Bei dem Microsoft SQL Server 2005 ist mit den enthaltenen Business Intelligence Funktionalitäten, besonders im Bereich der mittelständischen Unternehmen mit großem Wachstumspotential zu rechnen, da die Funktionalitäten für deren Anforderungen in geeignetem Umfang aufgebaut sind.

Eine bessere Qualität der Datenauswertung wird die Produktivität und Profitabilität des Mittelstands in Zukunft nachhaltig steigern.

---

<sup>221</sup> Hahne (2002a), S. 291

<sup>222</sup> Hahne (2002a), S. 292

<sup>223</sup> Hahne (1999), S. 168

## Glossar

**Anomalie:** Anomalien sind von der Realität abweichende Sachverhalte, die bei Einfüge-, Änderungs- und Löschoperationen auf einer Datenbank entstehen können.<sup>224</sup>

**Basisdatenbank:** Physische Datenbank, die eine integrierte Sicht auf (beliebige) Daten darstellt. Die Basisdatenbank dient nicht ausnahmslos einem speziellen Analyseanspruch und unterliegt deshalb nicht einem spezifischen Modellierungsansatz. Eine Aktualisierung und Modifikation des Datenbestandes ist möglich. Sie weist Ähnlichkeiten mit einer replizierten, föderierten Datenbank auf. Abhängig vom Verwendungszweck kann eine Historisierung stattfinden. In der Literatur wird die Basisdatenbank gelegentlich auch als Data Warehouse bezeichnet.<sup>225</sup>

**Data Warehouse:** Physische Datenbank, die eine integrierte Sicht auf (beliebige) Daten darstellt. Im Unterschied zur Basisdatenbank, steht der Auswertungsaspekt (analyse-orientiertes Schema) im Mittelpunkt, der sich oft in einem multidimensionalen Schema widerspiegelt. Häufig, aber nicht notwendigerweise, findet eine Historisierung der Daten statt, indem in periodischen Abständen Daten hinzugeladen, aber nicht modifiziert werden.<sup>226</sup>

**Datenbankdesign:** Der Prozess der Erstellung des Designs einer Datenbank, das die Operationen und Ziele des Unternehmens unterstützt.<sup>227</sup>

**Datenbankmanagementsystem:** Ein Datenbankmanagementsystem besteht aus einer Speicherungs- und einer Verwaltungskomponente. Die Speicherkomponente erlaubt, Daten und Beziehungen abzulegen, die Verwaltungskomponente stellt Funktionen und Sprachmittel zur Pflege und Verwaltung der Daten zur Verfügung.<sup>228</sup>

**Datenbankschema:** Unter einem relationalen Datenbankschema versteht man die formale Spezifikation der Datenbanken und Tabellen, unter der Angabe von Schlüssel- und Nichtschlüsselmerkmalen sowie von Integritätsbedingungen.<sup>229</sup>

**Datenmodell:** (engl. Data Model) Ein Datenmodell beschreibt auf strukturierte und formale Art die für ein Informationssystem notwendigen Daten und Datenbeziehungen.<sup>230</sup>

**Dimension:** Eine Dimension ist innerhalb des multidimensionalen Datenmodells eine ausgewählte Entität, mit der eine Analysesicht eines Anwendungsbereichs definiert

---

<sup>224</sup> Meier (1998), S. 181

<sup>225</sup> Bauer, Günzel (2001), S. 515

<sup>226</sup> Bauer, Günzel (2001), S. 516

<sup>227</sup> Conolly, Begg, Strachan (2002), S. 154

<sup>228</sup> Meier (1998), S. 182

<sup>229</sup> Meier (1998), S. 182

<sup>230</sup> Meier (1998), S. 182

wird. Dimensionen dienen der eindeutigen, orthogonalen Strukturierung des Datenraums.<sup>231</sup>

**Dimensionalität des Würfels:** Anzahl der Dimensionen, die einen Würfel aufspannen.<sup>232</sup>

**Entität:** Entitäten entsprechen Objekten der realen Welt unserer Vorstellung, sie werden durch die Merkmale charakterisiert und zu Entitätsmengen zusammengefasst.<sup>233</sup>

**Entitäten-Beziehungsmodell:** Das Entitäten-Beziehungsmodell ist ein Datenmodell, das Datenklassen (Entitätsmengen) und Beziehungen freilegt. Entitätsmengen werden grafisch durch Rechtecke, Beziehungsmengen durch Rhomben dargestellt.<sup>234</sup>

**Granularität:** Stufe des Verdichtungsgrades der Daten im Würfel; dabei haben Detaildaten den niedrigsten Verdichtungsgrad und zusammengefasste Daten (z.B. bei Aggregationen) einen höheren Verdichtungsgrad.<sup>235</sup>

**Integritätsbedingungen:** Integritätsbedingungen sind formale Spezifikationen über Schlüssel, Merkmale und Wertebereiche. Sie dienen dazu, die Widerspruchsfreiheit der Daten zu gewährleisten.<sup>236</sup>

**Kardinalitätsverhältnis:** Beschreibt die Anzahl von möglichen Beziehungen für jede teilnehmende Entität. (Kardinalitätsverhältnisse: 1:1; 1:M; M:N)<sup>237</sup>

**OLAP (Online Analytical Processing):** OLAP ist die explorative, interaktive Analyse auf Grundlage des konzeptuellen multidimensionalen Datenmodells.<sup>238</sup>

**Redundanz:** Die mehrfache Speicherung desselben Sachverhalts in einer Datenbank wird als Redundanz bezeichnet.<sup>239</sup>

**Relationenmodell:** Das Relationenmodell ist ein Datenmodell, das sowohl Daten als auch Datenbeziehungen in Form von Tabellen ausdrückt.<sup>240</sup>

**Schlüssel:** Ein Schlüssel ist eine minimale Merkmalskombination, die alle Tupel innerhalb einer Tabelle eindeutig identifiziert.<sup>241</sup>

---

<sup>231</sup> Bauer, Günzel (2001), S. 517

<sup>232</sup> Bauer, Günzel (2001), S. 517

<sup>233</sup> Meier (1998), S. 183

<sup>234</sup> Meier (1998), S. 183

<sup>235</sup> Bauer, Günzel (2001), S. 518

<sup>236</sup> Meier (1998), S. 183

<sup>237</sup> Connolly, Begg, Strachan (2002), S. 195

<sup>238</sup> Bauer, Günzel (2001), S. 519

<sup>239</sup> Meier (1998), S. 184

<sup>240</sup> Meier (1998), S. 184

<sup>241</sup> Meier (1998), S. 185

**SQL:** SQL (Structured Query Language) ist die wichtigste relationale Abfrage- und Manipulationssprache; sie wurde durch die ISO (International Organization for Standardization) genormt.<sup>242</sup>

**Würfel:** Mehrdimensionale Matrix, deren Zellen ein oder mehrere Kenngrößenwerte enthalten (z.B. Umsatz, Erlös). Der Würfel wird durch die Dimensionen (z.B. Produkt, Kunde, Zeit) als Achsen mit ihren jeweiligen Ausprägungen (eigentlich Datenquader) aufgespannt.<sup>243</sup> Der Begriff des Würfels kommt von der Vorstellung des mehrdimensionalen Datenraums als ein Quader aufgespannt von den Dimensionen mit den Beschriftungen der Dimensionselemente, deren Zellen die Datenwerte beinhalten, nahe. Obwohl nicht alle Dimensionen die gleiche Anzahl an Elementen aufweisen, wird trotzdem der Begriff Würfel statt Quader benutzt, da sich dieser bereits allgemein durchgesetzt hat.<sup>244</sup>

**Würfelschema:** Das Schema eines Würfels wird durch Dimensionen und Kenngrößen bestimmt. Ein Würfel entsteht durch Instanziierungen eines multidimensionalen Schemas.<sup>245</sup>

**Würfelzelle:** Kleinster Teil eines Würfels, der durch die Dimensionselemente adressiert werden kann.<sup>246</sup>

---

<sup>242</sup> Meier (1998), S. 185

<sup>243</sup> Bauer, Günzel (2001), S. 520

<sup>244</sup> Hahne (2002a), S. 83

<sup>245</sup> Bauer, Günzel (2001), S. 520

<sup>246</sup> Bauer, Günzel (2001), S. 520

## Literaturverzeichnis

**Bauer, A.; Günzel, H.** (Hrsg.) (2001): Data Warehouse Systeme. Architektur, Entwicklung, Anwendungen, Heidelberg

**Behme, W.; Holthuis, J.; Mucksch, H.** (2000): Umsetzung multidimensionaler Strukturen, In: Mucksch, H.; Behme, W. (Hrsg.) (2000): Das Data Warehouse - Konzept, Architektur – Datenmodelle – Anwendungen, 4. Auflage, Wiesbaden

**Chamoni, P.; Gluchowski, P.** (1999): Analytische Informationssysteme – Einordnung und Überblick, In: Chamoni, P; Gluchowski, P. (Hrsg.) (1999): Analytische Informationssysteme: Data Warehouse, On-Line Analytical Processing, Data Mining, 2. Auflage, Berlin

**Chamoni, P.; Gluchowski, P.** (1998): Analytische Informationssysteme – Einordnung und Überblick, In: Chamoni, P; Gluchowski, P. (Hrsg.) (1998): Analytische Informationssysteme: Data Warehouse, On-Line Analytical Processing, Data Mining, Berlin

**Chamoni, P.; Gluchowski, P.** (2000): On-Line Analytical Processing (OLAP), In: Mucksch, H.; Behme, W. (Hrsg.) (2000): Das Data Warehouse - Konzept, Architektur – Datenmodelle – Anwendungen, 4. Auflage, Wiesbaden

**Chen, P. P. S.; Knöll, H.-D.** (1991): Der Entity-Relationship-Ansatz zum logischen Systementwurf: Datenbank und Programmentwurf, Mannheim

**Connolly, T.; Begg, C.; Strachan, A.** (2002): Datenbanksysteme – Eine praktische Anleitung zu Design, Implementierung und Management, München

**Date, C.** (1986): Relational Database: Selected Writings, Massachusetts

**Gabriel, R.; Röhrs, H.-P.** (2003): Gestaltung und Einsatz von Datenbanksystemen – Data Base Engineering und Datenbankarchitekturen, Berlin

**Garcia, M. F.; Reding, J.; Whalen, E.; DeLuca, S.** (2000): Microsoft SQL Server 2000 – Das Handbuch

**Gluchowski, P.; Chamoni, P.** (1999): Analytische Informationssysteme: Data Warehouse – On-line Analytical Processing – Data Mining, 2. Auflage, Berlin

**Hahne, M.** (2004a): Grafische Repräsentation mehrdimensionaler Datenmodelle des SAP Business Information Warehouse, Adresse:

[http://www.cundus.de/downloads/MKW12004\\_Hahne\\_paper.pdf](http://www.cundus.de/downloads/MKW12004_Hahne_paper.pdf), letzter Zugriff:

28.07.2004

**Hahne, M.** (2004b): Modellierung für BI-Systeme – Mehrdimensionales Datenmodell der Microsoft SQL Server 2000 Analysis Services™, Adresse:

[http://www.cundus.de/downloads/TDW12004\\_Hahne\\_slides.pdf](http://www.cundus.de/downloads/TDW12004_Hahne_slides.pdf), letzter Zugriff:

28.07.2004

**Hahne, M.** (2003): Time aspects in SAP Business Information Warehouse, Adresse:

[http://www.cundus.de/downloads/CE03\\_Hahne\\_papers.pdf](http://www.cundus.de/downloads/CE03_Hahne_papers.pdf), letzter Zugriff: 28.07.2004

**Hahne, M.** (2002a): Logische Modellierung mehrdimensionaler Datenbanksysteme, Dissertation, Universität Bochum, Wiesbaden

**Hahne, M.** (2002b): Transformation mehrdimensionaler Datenmodelle. In: von Maur, E.; Winter, R. (Hrsg.): Vom Data Warehouse zum Corporate Knowledge Center, Physica-Verlag, Heidelberg, 2002.

**Hahne, M.** (1999): Logische Datenmodellierung für das Data Warehouse - Bestandteile und Varianten des Star Schemas, In: Chamoni, P; Gluchowski, P. (Hrsg.) (1999): Analytische Informationssysteme: Data Warehouse, On-Line Analytical Processing, Data Mining, 2. Auflage, Berlin

**Holthuis, J.** (2000): Grundüberlegungen für die Modellierung einer Data Warehouse-Datenbasis, In: Mucksch, H.; Behme, W. (Hrsg.) (2000): Das Data Warehouse - Konzept, Architektur – Datenmodelle – Anwendungen, 4. Auflage, Wiesbaden

**Holthuis, J.** (2001): Der Aufbau von Data Warehouse-System: Konzeption-Datenmodellierung-Vorgehen, 2. Auflage, Wiesbaden

**Jacobson, R.** (2000): SQL Server 2000 Analysis Services – Schritt für Schritt

**Kaiser, E.** (1992): Semantische Datenmodellierung in Theorie und Praxis, Inauguraldissertation, Universität Mannheim, Lehrstuhl für Allgemeine Betriebswirtschaftslehre, Organisation und Wirtschaftsinformatik, Mannheim

**Kimball, R.; Ross, M.** (2002): The Data Warehouse Toolkit, 2<sup>nd</sup> edition, New York

**Larsen, D.** (2000): Data Transformation Services (DTS) in Microsoft SQL Server 2000, Adresse:

[http://msdn.microsoft.com/SQL/sqlwarehouse/DTS/default.aspx?pull=/library/en-us/dnsql2k/html/dts\\_overview.asp](http://msdn.microsoft.com/SQL/sqlwarehouse/DTS/default.aspx?pull=/library/en-us/dnsql2k/html/dts_overview.asp), letzter Zugriff: 01.08.2004

**Lehmann, P.** (2001): Meta-Datenmanagement in Data-Warehouse-Systemen. Rekonstruierte Fachbegriffe als Grundlage einer Konstruktiven, konzeptionellen Modellierung; Dissertation, Otto-von-Guericke-Universität Magdeburg, Fakultät für Informatik, Institut für Technische und Betriebliche Informationssysteme, Arbeitsgruppe Wirtschaftsinformatik

**Lehner, F.** (1995a): Grundfragen und Positionierung der Wirtschaftsinformatik, In: Lehner, F.; Hildebrand, K.; Maier, R. (1995): Wirtschaftsinformatik: Theoretische Grundlagen, München

**Lehner, F.** (1995b): Modelle und Modellierung, In: Lehner, F.; Hildebrand, K.; Maier, R. (1995): Wirtschaftsinformatik: Theoretische Grundlagen, München

**Lehner, W.** (2003): Datenbanktechnologie für Data-Warehouse-Systeme: Konzepte und Methoden, Heidelberg

**Lehner, F.; Hildebrand, K.; Maier, R.** (1995): Wirtschaftsinformatik: Theoretische Grundlagen, München

**Matthiessen, G.; Unterstein, M.** (2000): Relationale Datenbanken und SQL – Konzepte der Entwicklung und Anwendung, München

**Meier, A.** (1998): Relationale Datenbanken – Eine Einführung für die Praxis, 3. Auflage, Berlin

**Mertens, P.; et.al.** (2001): Lexikon der Wirtschaftsinformatik, 4. Auflage, Berlin

**Microsoft** (2004a): Business Intelligence and Data Warehousing in SQL Server 2005, Adresse:

<http://www.microsoft.com/technet/prodtechnol/sql/2005/evaluate/dwsqlsy.msp>, letzter Zugriff: 29.07.2004

**Microsoft** (2004b): Warum SQL Server 2000 Reporting Services?, Adresse:

<http://www.microsoft.com/germany/ms/sql/2000/reportingservices/warum.htm>, letzter Zugriff: 13.07.2004

**Microsoft** (2004c): Supported Data Types and Data Type Mappings, Adresse:

[http://msdn.microsoft.com/library/default.asp?url=/library/en-us/sqlce/html/lce\\_accessing\\_data.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/sqlce/html/lce_accessing_data.asp), letzter Zugriff: 15.08.2004

**Mucksch, H.** (1999): Das Data Warehouse als Datenbasis analytischer Informationssysteme – Architektur und Komponenten, In: Chamoni, P; Gluchowski, P. (Hrsg.)

(1999): Analytische Informationssysteme: Data Warehouse, On-Line Analytical Processing, Data Mining, 2. Auflage, Berlin

**Mucksch, H.; Behme, W.** (2000): Das Data Warehouse-Konzept als Basis einer unternehmensweiten Informationslogistik, In: Mucksch, H.; Behme, W. (Hrsg.) (2000):

Das Data Warehouse - Konzept, Architektur – Datenmodelle – Anwendungen, 4. Auflage, Wiesbaden

**Mucksch, H.; Behme, W.** (2000): Das Data Warehouse - Konzept, Architektur – Datenmodelle – Anwendungen, 4. Auflage, Wiesbaden

**Nunn, M.** (2004): An Overview of SQL Server 2005 for the Database Developer, Adresse:

[http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnsq90/html/sql\\_ovvyukonde.asp?frame=true](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnsq90/html/sql_ovvyukonde.asp?frame=true), letzter Zugriff: 29.07.2004



**o.V.** (2004): Microsoft SQL Server 2005 Beta 2 am Start, Adresse:  
<http://entwickler.com/itr/news/psecom,id,16821,nodeid,30.html>, letzter Zugriff:  
29.07.2004

**o.V.** (2003): Tools für SQL Server 2000 und SQL Server Yukon, Adresse:  
<http://www.winfuture.de/news,12160.html>, letzter Zugriff: 29.07.2004

**Schinker, H. ; Bange, C.** (1999): Werkzeuge zum Aufbau analytischer Informationssysteme – Marktübersicht, In: Chameni, P; Gluchowski, P. (Hrsg.) (1999): Analytische Informationssysteme: Data Warehouse, On-Line Analytical Processing, Data Mining, 2. Auflage, Berlin

**Schinker, H.; Bange, C.; Mertens, H.** (1999): Data Warehouse und Data Mining: Marktführende Produkte im Vergleich, 2. Auflage, München

**Schlageter, G.; Stucky, W.** (1983): Datenbanksysteme: Konzepte und Modelle, 2. Auflage, Stuttgart

**Schmidt-Thieme, L.** (2002): E-Business: Datenbanken / Data Warehousing, Adresse:  
<http://www.informatik.uni-freiburg.de/cqnm/lehre/eb-03s/eb9.pdf>, letzter Zugriff:  
21.08.2004

**Schreier, U.** (2001): Entity-Relationship-Darstellung. In: Mertens, P.; et.al.: Lexikon der Wirtschaftsinformatik, 4. Auflage, Berlin

**Totok, A.** (2000a): Modellierung von OLAP- und Data-Warehouse-Systemen, Dissertation, Technische Universität Braunschweig, Wiesbaden

**Totok, A.** (2000b): Grafische Notationen für die semantische multidimensionale Modellierung, In: Mucksch, H.; Behme, W. (Hrsg.) (2000): Das Data Warehouse - Konzept, Architektur – Datenmodelle – Anwendungen, 4. Auflage, Wiesbaden

**Vossen, G.** (1999): Datenbankmodelle, Datenbanksprachen und Datenbankmanagement-Systeme, 3. Auflage, München

**Wedekind, H.** (2001): Datenmodell, In: Mertens, P.; et.al.: Lexikon der Wirtschaftsinformatik, 4. Auflage, Berlin

**Wieken, J.-H.** (2002): Der Weg zum Data Warehouse – Wettbewerbsvorteile durch strukturierte Unternehmensinformationen, München

## Erklärung

Hiermit erkläre ich, dass ich die vorliegende Diplomarbeit selbständig angefertigt habe. Es wurden nur die in der Arbeit ausdrücklich benannten Quellen und Hilfsmittel benutzt. Wörtlich oder sinngemäß übernommenes Gedankengut habe ich als solches kenntlich gemacht.

---

Stuttgart, 27. August 2004

A handwritten signature in black ink, consisting of a stylized 'H.' followed by a series of loops and a long horizontal stroke.

---

Unterschrift